

(19)世界知的所有権機関
国際事務局(43)国際公開日
2004年3月25日 (25.03.2004)

PCT

(10)国際公開番号
WO 2004/025651 A1

(51) 国際特許分類⁷: G11B 27/00, 27/10, 20/10, 20/12 (71) 出願人(米国を除く全ての指定国について): 松下電器産業株式会社 (MATSUSHITA ELECTRIC INDUSTRIAL CO., LTD.) [JP/JP]; 〒571-8501 大阪府門真市大字門真 1006 番地 Osaka (JP).

(21) 国際出願番号: PCT/JP2003/011679

(22) 国際出願日: 2003年9月12日 (12.09.2003) (72) 発明者; および

(25) 国際出願の言語: 日本語 (75) 発明者/出願人(米国についてのみ): 岡田智之 (OKADA,Tomoyuki) [JP/JP]; 〒631-0078 奈良県奈良市富雄元町 1-8-19-303 Nara (JP). 小塙雅之 (KOZUKA,Masayuki) [JP/US]; 91007 カリフォルニア州アーティア、サウスゴールデンウエストアベニュー 825、5号室 CA (US). 中村和彦 (NAKAMURA,Kazuhiko) [JP/JP]; 〒573-0084 大阪府枚方市香里ヶ丘 11-35-53 Osaka (JP). 上坂靖 (UESAKA,Yasushi) [JP/JP]; 〒669-1348 兵庫県三田

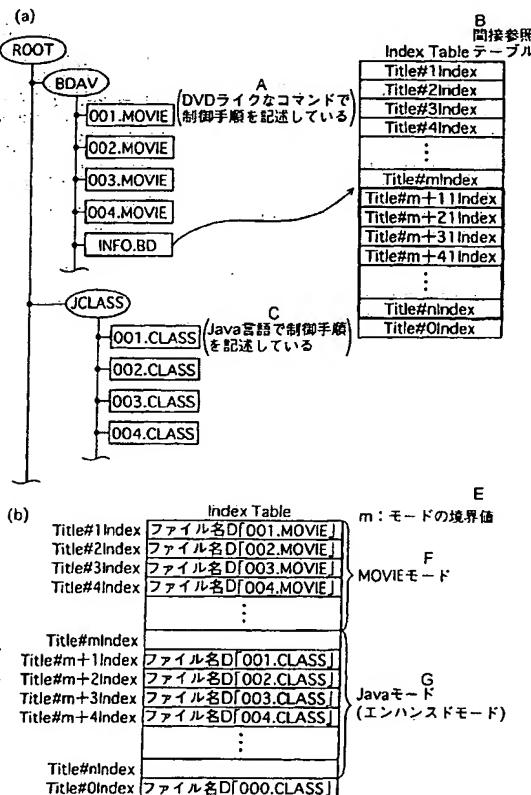
(26) 国際公開の言語: 日本語

(30) 優先権データ:
60/409,999 2002年9月12日 (12.09.2002) US
60/440,623 2003年1月17日 (17.01.2003) US

[続葉有]

(54) Title: RECORDING MEDIUM, REPRODUCTION DEVICE, PROGRAM, REPRODUCTION METHOD, AND RECORDING METHOD

(54) 発明の名称: 記録媒体、再生装置、プログラム、再生方法、記録方法



(57) Abstract: An AV stream obtained by multiplexing a video stream and an audio stream is recorded on a recording medium (BD-ROM). A MOVIE object is a scenario indicating a reproduction procedure of the moving picture data described by using a command for a reproduction device. In addition to this MOVIE object, an enhanced mode scenario (Java object, WebPage object) is recorded. This scenario is described by Java language and markup language and indicates a control procedure for the reproduction device. The Java object and the WebPage object can inherit the register set value having a value set by the MOVIE object and quote a part of the moving picture data reproduced by the MOVIE object.

(57) 要約: 記録媒体(BD-ROM)には、ビデオストリーム、オーディオストリームを多重化することにより得られたAVストリームが記録されている。MOVIEオブジェクトは、再生装置向けのコマンドを用いて記述された動画データの再生手順を示すシナリオである。このMOVIEオブジェクトとは別に、エンハンスドモードのシナリオ(Javaオブジェクト、WebPageオブジェクト)が記録されている。このシナリオには、Java言語、マークアップ言語で記述されており、再生装置に対する制御手順を示す。Javaオブジェクト、WebPageオブジェクトは、MOVIEオブジェクトにより値が設定されたレジスタ設定値を引き継いだり、MOVIEオブジェクトにて再生された動画データの一部を引用することができる。

A...CONTROL PROCEDURE IS DESCRIBED BY DVD-LIKE COMMAND

B...INDIRECT REFERENCE TABLE

C...CONTROL PROCEDURE IS DESCRIBED BY Java LANGUAGE

D...FILE NAME

E...m:MODE BOUNDARY VALUE

F...MOVIE MODE

G...Java MODE (ENHANCED MODE)

WO 2004/025651 A1



市つつじが丘北 2-16-16 Hyogo (JP). 池田 航
(IKEADA,Wataru) [JP/JP]; 〒534-0023 大阪府 大阪市
都島区都島南通 2-1-3-1205 Osaka (JP).

(74) 代理人: 中島 司朗 (NAKAJIMA,Shiro); 〒531-0072 大
阪府・大阪市北区豊崎三丁目 2番 1号 淀川5番館 6F
Osaka (JP).

(81) 指定国(国内): AE, AG, AL, AM, AT, AU, AZ, BA, BB,
BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK,
DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR,
HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR,
LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ,
NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE,
SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US,
UZ, VC, VN, YU, ZA, ZM, ZW.

(84) 指定国(広域): ARIPO 特許 (GH, GM, KE, LS, MW, MZ,
SD, SL, SZ, TZ, UG, ZM, ZW), ユーラシア特許 (AM,
AZ, BY, KG, KZ, MD, RU, TJ, TM), ヨーロッパ特許
(AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB,
GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR),
OAPI 特許 (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW,
ML, MR, NE, SN, TD, TG).

添付公開書類:
— 國際調査報告書

2文字コード及び他の略語については、定期発行される
各PCTガゼットの巻頭に掲載されている「コードと略語
のガイダンスノート」を参照。

明細書

記録媒体、再生装置、プログラム、再生方法、記録方法

技術分野

5 本発明は、Blu-Ray Disc Read Only Memory(以下 BD-ROM と略す)等の動画データ用の記録媒体、記録媒体の再生装置、プログラム、再生方法、記録方法に関し、記録媒体により映画作品等を頒布する技術に関する。

背景技術

10 記録媒体による映画作品の頒布は、映画制作者、供給者にとって大きな収入源である。映画作品の動画データは、著名な俳優を出演させ、多大な制作費をかけて作成されているので多大な財産価値をもつ。映画作品の頒布にあたっては、記録媒体や再生装置の機能を利用して、動画データの付加価値を高めることが、商品戦略上極めて有意義とされている。

15 DVD における付加価値の高め方としては、動画データを記録した DVD に、その動画データを利用したクイズやゲームを記録しておき、ユーザに遊んでもらおうというものがある。同じ動画データを使っていたとしても、映画として動画データを見るのと、ゲームの一場面として見るとではユーザが受ける印象はガラリと変わる。僅かな労力で、ユーザに新鮮な印象を与えつづけることができるのと、かかる高付加価値化は、映画制作者にとって強力な武器となる。

20

DVD における付加価値の高め方には、以下の特許文献 1 に記載された先行技術がある。

{特許文献 1} 特許第 2813245 号

25

ところで従来の高付加価値化は、DVD の再生装置が描画可能な副映像や解釈可能なコマンドを用いて実現されていた。つまりこれらの高付加価値化は、再生装置機能を応用した結果の副産物といえる。いまや家電機器のネット家電化の進行に伴い、この再生装置機能は劇的な進化を遂げようとしている。ネット家電化にあたって、再生装置を始めとする家

電機器は、ブラウザや Java 仮想マシンを標準装備し、ブラウザによる動作モードや Java 仮想マシンによる動作モードを具備をすることとなり、ネットワークを通じた様々なサービスをユーザに供給することができる。こうした再生装置側の進化を踏まえ、映画供給者は、Java 仮想
5 マシンやブラウザの特性を活かして動画データ本体の付加価値を高めるという新たなコンテンツ作りを機器メーカーに提案していくと考えられる。しかし java 仮想マシンやブラウザは広く普及しているとはいえ、これらを用いて動画データ本体の付加価値を高めるという考え方には、従来には一切存在しなかった。かかる要望があったとしても、実際の民生機器
10 のマーケットでは、Java 仮想マシン、ブラウザをもった再生装置やもっていない再生装置の両者が出現すると考えられる。Java 仮想マシン、ブラウザをもたない再生装置に、本記録媒体が装填された場合に、動作保証がなされないのならば、付加価値を高めた記録媒体は市場から排斥される恐れがある。
15 更に、ネット家電化された再生装置向けに映画作品が記録された記録媒体を発布する場合、映画作品の著作権は未知数のリスクに晒されることになる。Java 仮想マシン、ブラウザを用いた動作モードに魅力があるにせよ、著作権者は大きな不安を抱かずにはいられない。

発明の開示

20 本発明の目的は、再生装置に Java 仮想マシンやブラウザが装備されている場合に、これらを利用して動画データ本体の付加価値を高めつつも、様々なタイプの再生装置での動作保証を実現することができる記録媒体及び再生装置を提供することである。

上記目的は、動画データ、複数のプログラム、テーブルが記録された記録媒体であって、各プログラムは、動画データの再生制御手順を示し
25 テーブルは、記録媒体における各プログラムの識別情報(1)、各プログラムがムービーモード及びエンハンスドモードの何れに属するかを示す情報(2)を含んでおり、何れか1つのプログラムは、分岐コマンドを含み、前記分岐コマンドは、テーブルを介した間接参照の形式で、分岐先
30 を指定している記録媒体により達成される。ムービーモードにおける映

画としての動画データの再生と、エンハンスドモードにおける制御とを、分岐コマンドを用いて動的に切り換えることができる。エンハンスドモードにおける制御が、Java仮想マシンやブラウザを用いたものならば、記録媒体に記録された動画データは、通常の映画を再生させるための画面、Java仮想マシンやブラウザによる画面のどちらかで再生されることになる。かような画面切り換えで、従来にないような、斬新な視覚効果が実現されることになる。

動的なモード切り換えを活かしたゲーム等を再生装置に実行させることができるので、動画データ本体の付加価値を高めることができる。またムービーモードからエンハンスドモードへの分岐は、テーブルを介した間接参照にて分岐先を指定している。テーブルの記述内容を工夫すれば、Java仮想マシン、ブラウザをもつ再生装置への装填時と、もたない再生装置への装填時とで、分岐先を変えるという動作も実現可能になる。かかる分岐先の変化により、Java仮想マシン、ブラウザをもたない再生装置への装填時に、エンハンスドモードプログラムへの分岐の途を閉ざすことができ、あらゆるタイプの再生装置での動作保証を実現することができる。

ここで前記テーブルは、複数プログラムのそれぞれに対応した複数のインデックスを含み、各インデックスは、対応するプログラムがエンハンスドモードに属するか、ムービーモードに属するかを示しており、

前記分岐コマンドにおける間接参照の形式とは、インデックスについてのラベルを用いて、分岐先プログラムを指定していくてもよい。

プログラムからプログラムへの分岐は、インデックスを参照してなされる。このインデックスは、プログラムをどのモードで実行せねばよいかが示されているので、モードを切り換えたうえで、他のプログラムに分岐するという、モード切り換えを伴った分岐を再生装置に容易に実行させることができる。

ここで前記テーブルにおけるインデックスには、予備のインデックス

があり、予備のインデックスは、ムービーモードしか実行できない再生装置において、エンハンスドモードのプログラムへの分岐が命じられた場合に、エンハンスドモードのプログラムの代わりとなるムービーモードプログラムと対応していてもよい。

5 エンハンスドモードがない再生装置に装填された場合、予備のインデックスを参照して分岐を行うことで、プログラムの分岐時において、ムービーモードのプログラムへと誘導することができる。エンハンスドモードのプログラムへの分岐の途を閉ざすことができるので、意図しない誤動作を防ぐことができる。

10 前記各ムービーモードのプログラム及びエンハンスドモードのプログラムのそれぞれは、2以上の実行モジュールのそれぞれにより実行され、2以上の実行モジュールは、共通の制御階層上のレジデントプログラムであり、

15 前記再生制御手順は、当該制御階層から供給される関数を用いて記述されていてもよい。

ここで前記制御関数から供給される関数は、

予め定められた再生経路に基づく再生制御を再生装置に行わせる関数

(1)、

再生装置におけるレジスタに所定の値を設定する関数(2)、

20 再生装置におけるレジスタの設定値を取得する関数(3)のうち何れかであってもよい。

ムービーモードにおいて設定されたレジスタの値を引き継ぎ、その設定値に応じて、異なる手順を実行するという制御を、エンハンスドモードで実現することができる。例えば、エンハンスドモードプログラムが、キャラクタを画面上で動作させるという手順である場合、ムービーモードからエンハンスドモードにレジスタの設定値を引き継がせることで、動画データに対するユーザ設定と、コンピュータ・グラフィックスのキャラクタの動作とが密接に関連しあうような制御手順を再生装置に実行させることができる。かかる制御手順を取り入れることで、映画作品を制作するにあたっての表現の幅が広がり、再生制御の記述という僅かな

投資で、動画データの付加価値を効果的に高めることができ、映画制作者、配給者の有力な武器となる。

ゲームの制作により、付加価値の顕著な向上を図るのなら、前記エンハンスモードは、仮想マシンに、プログラムを実行させるモードであり、
5 エンハンスドモードのプログラムは、仮想マシン向けプログラミング言語により記述すればよいか

モード切り換えを伴った分岐を実現することで、Java言語(登録商標)の開発環境を創設することにより、多くのソフトハウスの参入を促すことができる。これにより、Javaモードへの切り換えを伴いながら、
10 動画データを再生させてゆくという斬新なスタイルの映像作品が多く制作されることになり、映画作品の市場を活性化させることができる。

DVD再生環境との融合を考えるのなら、前記ムービーモードのプログラムは、ボタンコマンドを含みボタンコマンドは、エンハンスドモードのプログラムへと分岐するコマンドであり、ボタンコマンドは、動画データ、副映像データと多重化されて多重化ストリームとして記録媒体に記録され、副映像データは、ボタンの画像データであり、前記ボタンコマンドは、ボタンの画像データに対して確定操作がなされた際、実行されるようにするのがよい。

ボタンコマンドは、DVD再生装置が解釈可能なコマンドに準拠したものであり、これを用いて制御手順を記述することで、DVDとの制御手順の互換性を図ることができる。かかる互換性の確保により、1つの映画作品を、DVD、BD-ROMの両者でリリースする場合に、制御構造の共通化が図れ、BD-ROMによる映画作品のリリースに際してもオーサリングの手間を軽減することができる。

25 またDVDライクなムービーモードから、Java仮想マシン、ブラウザによるエンハンスドモードに分岐するというモード移行も可能になり。DVD機能、Java仮想マシン／ブラウザ機能と交えた再生制御が可能になる。

30 ブラウザを用いるとしても、一歩間違えれば、映画作品を誹謗中傷するサイトと共に、映画作品が再生されることもある。かような再生を認

めれば、映画作品の品位に泥を塗ってしまう恐れがある。自らが手掛けた映画の品位が汚されることとは、供給会社や制作会社にとって堪え難いことであり、機器メーカとの関係に、軋轢を生む要因になりかねない。

かのような軋轢を避けるには、エンハンスドモードプログラムを、マークアップ言語で記述することが望ましい。

マークアップ言語で制御手順を記述するので、Web サイトからの情報を交えた制御手順により、動画データの付加価値を高めることができる。どの Web サイトからの情報を交えるかは、制作者サイドが決定するため、インターネットを介した不特定多数からの誹謗や中傷に映画作品が晒さられることが避けられる。

テーブルを介した間接参照にてモード遷移を行うことの技術的意義は上述したような動作保証に留まらない。つまり何等かの条件が再生装置側で満たされた場合、ムービーモードからエンハンスドモードへの遷移の途を閉ざすことができるので、以下のような事情が再生装置にある場合に有意義である。

映画作品の著作権保護のための暗号鍵がある再生装置で暴露され、再生装置が鍵管理センターにより無効化された場合(1)、

ユーザがリッパソフトを利用して記録媒体に記録された映画作品を不正にコピーしたため、映画作品の複製物がネットワーク上に或はされる恐れがある場合(2)、

エンハンスドモード実行には追加料金の支払いがいるにも拘らず、追加料金が支払われない場合(3)、

再生装置のシステムに障害が発生したため、ネットワークから遮断したい場合(4)、

Java 仮想マシンやブラウザにバージョンコンフリクションがある場合(5)、

個人情報の漏洩やウィルスソフトの感染の可能性があり、再生装置をネットワークから遮断したい場合(6)、

ネットワークから記録媒体の記録内容を読み取ろうとする不正機器から、記録媒体の映画作品を保護するため、再生装置をネットワークから

遮断したい場合(7)、

といった(1)～(7)の事情が再生装置に存在する場合にモード遷移を閉ざすことも可能である。

たとえ再生装置が常時接続可能な環境におかれていたとしても、再生装置の動作や動画データの著作権保護が保証されない場合、ユーザに債務不履行がある場合、Java仮想マシンやブラウザへの遷移を行わないようにムービーモードプログラムを記述することができるので、ネット家電化された再生装置向けに映画作品を頒布する場合の著作権者の不安を払拭することができる。

更に上述したような間接参照による分岐コマンドは、後日の差し替えを僅かな手間で実現できる点において、映画作品の頒布を行う業者にとって大きなメリットになる。つまり記録媒体に記録された特定の動画データに道徳・倫理的な問題がある場合、テーブルと、差し替え映像とを再生装置にダウンロードさせれば、新たなテーブルを介した間接参照を再生時に再生装置に行わせることで、問題がある映像の代わりに、差し替え映像をユーザに視聴させることができる。一部差し替えにあたって記録媒体に記録されたプログラムを一切書き換えなくてもよいので、かかる問題が生じたとしても、記録媒体を回収するというリスクを避けることができる。

また差し替え映像でなくとも、記録媒体に記録されている複数動画データのうち、特定のものを再生させたくない場合、記録媒体に記録されている動画データの順序を入れ替えたい場合も、テーブルのダウンロードさえ行えば、記録媒体側のプログラムを書き換えなくてもよいので、問題発覚時の業者側の負担を軽減することができる。

また本発明に係る記録媒体の最も広い上位概念は、以下のようなものであってもよい。

つまり、動画データ、プログラムが記録された記録媒体であり、前記プログラムは、動画データの再生制御手順を示し

再生制御手順は、動画データの再生中において、再生装置で発生するイベントをトリガにして実行され、

前記イベントとは、

現在の再生位置が、動画データの再生時間軸における所定の時点に到達したことを示すイベント(i)、

動画データの再生時間軸において、所定の時間間隔だけ、現在の再生位置が進行したことを示すイベント(ii)の何れかであってもよい。

また本発明に係る再生装置の最上位概念は、以下のようなものであってもよい。つまり、動画データ及びプログラムが記録された記録媒体についての再生装置であって、動画データの再生、及び、再生進行に同期したイベント発生を行う再生制御エンジンと、

再生制御エンジンがイベントを発生すると、イベントハンドラを実行する実行モジュールとを備え、

前記イベントとは、

現在の再生位置が、動画データの再生時間軸における所定の時点に到達したことを示すイベント(i)、

動画データの再生時間軸において、所定の時間間隔だけ、現在の再生位置が進行したことを示すイベント(ii)の何れかであってもよい。

動画データの再生と同期して動作するようなプログラムを、イベントハンドラとして作成することができるので、プログラマによる開発を容易にことができる。

20 図面の簡単な説明

図1は、本発明に係る記録媒体の、使用行為についての形態を示す図である。

図2は、BD-ROMの構成を示す図である。

図3は、ディレクトリ構造を用いてBD-ROMの応用層フォーマットを表現した図である。

図4は、機能的な観点から、ファイルを分類した場合の分類図である。

図5は、BD-ROMが対象としているソフトウェアのレイアモデルを示す図である。

図6は、AVストリームがどのように構成されているかを模式的に示す図である。

図7はAVストリームがどのようにBD-ROMに記録されるかを模式的に示す図である。

図8は、ストリーム管理情報の内部構成を示す図である。

図9は、PL情報の内部構成を示す図である。

5 図10は、PL情報による間接参照を模式化した図である。

図11は、図10に示したPLとは、別のPLを定義する場合の一例を示す図である。

図12は、レイヤモデルの第4層(動的シナリオ)における再生モードを示す図である。

10 図13は、3つのモードの動的な再生制御にて作成される映画作品を示す図である。

図14はNavigation Button情報の構成を示す図である。

図15は、AVストリームに多重化されているNavigation Button情報でボタンの制御を行う場合の一例を示す図である。

15 図16は、AVストリーム中のNavigation Button情報が図15のように設定されている場合のMOVIEオブジェクトの一例を示す。

図17は、図16のMOVIEオブジェクトのコマンドにより分岐がなされた場合のJavaオブジェクトの記述例である。

20 図18は、MOVIEオブジェクトとJavaオブジェクトとの関連を示す図である。

図19は、MOVIEオブジェクトにおけるキャラクタ紹介と、Javaモードにおけるゲームとから構成されるタイトルを模式的に示す図である。

図20は、JavaオブジェクトからのPL再生を模式的に示した図である。

25 図21は、MOVIEモードで使用されたいいたPLとは別のPLを用いることによる、シーン引用を示す図である。

図22は、WebResponseオブジェクトの記述例である。

図23(a)は、INFO.BD-ROMの内部構成を示す図である。

図23(b)は、Index TableのIndexを示す図である。

30 図24(a)は、複数の動的シナリオ

(001.MOVIE, 002.MOVIE, 003.MOVIE……、

001.CLASS, 002.CLASS, 003.CLASS……)が記録されている BD-ROM を示す図である。

図 24 (b) は、BD-ROM に図 24 (a) に示すような複数動的シナリオが記述されている場合の Index Table の記述例を示す図である。

図 25 (a) は、図 24 (b) のように Index Table が記述された場合のフルシステムにおける間接参照を示す図である。

図 25 (b) は、コアシステムにおける間接参照を示す図である。

図 26 は、図 18 に示した MOVIE オブジェクトから Java オブジェクトへの分岐がどのように行われるかを模式的に示す図である。

図 27 は、コアシステムの再生装置に図 18 に示したシナリオが記録された BD-ROM が装填された場合、どのような分岐が行われるかを示す図である。

図 28 は、本発明に係る再生装置の内部構成を示す図である。

図 29 は、モジュールマネージャ 20 による処理手順を示すフローチャートである。

図 30 は、モジュールマネージャ 20 による処理手順を示すフローチャートである。

図 31 は、再生制御エンジン 12 による PLPlay 関数の実行手順を示すフローチャートである。

図 32 は、第 2 実施形態に係る BD-ROM のファイル構成を示す図である。

図 33 は、PLMark 及び ClipMark の共通構成を示す図である。

図 34 (a) (b) は、プレイリスト #1 の再生中に出現する TimeEvent を定義する場合の PLMark の記述例を示す図である。

図 35 は、プレイリスト #1 の再生中において UserEvent を定義する場合の PLMark の記述例を示す図である。

図 36 は、TimeEvent により駆動されるハンドラの一例を示す図である。

図 37 は UserEvent により駆動されるハンドラの一例を示す図である。

図38は、図36、図37のプレイリストにて再生されるピクチャと、イベントハンドラにより描画されるイメージとを合成して、イベントハンドラによる条件分岐を描いた図である。

図39は、本実施形態に係る再生制御エンジン12の処理手順を示す
5 図である。

図40は、Backward Eventの技術的意義を説明するための模式図である。

図41(a)は、Javaモードにおいて、PL再生が開始した時点の表示画面の一例である。

10 図41(b)は、PLの再生時点が時刻t1に到達した場合の表示画面の一例である。

図41(c)は、PLの再生時点が時刻t2に到達した場合の表示画面の一例である。

15 図42(a)は、巻戻しによりPLの再生時点が時刻t1に到達した場合の表示画面の一例である。

図42(b)は、巻戻しによりPLの再生時点が時刻t0に到達した場合の表示画面の一例である。

図42(c)は、PLの再生時点が時刻t1に再到達した場合の表示画面の一例である。

20 図43は、早送り時のTime Event発生を模式的に示す図である。

図44(a)は、Javaモードにおいて、PL再生が開始した時点の表示画面の一例である。

図44(b)は、PLの再生時点が時刻t1に到達した場合の表示画面の一例である。

25 図44(c)は、時刻t2への到達時に、早送りを行った表示画面の一例である。

図45は、Pause Eventの発生を模式的に描いた図である。

図46は、BD-ROMにより実現されるメニュー階層を示す図である。

図47は、階層をもったメニューを動作させるためのMOVIEオブジェ
30 クトを示す図である。

図48は、第4実施形態に係るモジュールマネージャ20の処理手順を示す図である。

図49は、Javaモードに属するクラスのメンバー関数を示す図である。

5 図50は、Javaモード内のメンバー関数を介した制御を示す図である。

図51は、第6実施形態に係るメニュー階層を示す図である。

図52は、第6実施形態に係るMOVIEオブジェクト、Index Tableを示す図である。

10 図53は、スケジュールテーブルの一例を示す図である。

図54は、第7実施形態に係るPL情報の構成を示す図である。

図55は、デジタルストリーム本体をMOVIEモードと、Javaモードとでシェアリングしつつ、PL情報をMOVIEモードと、Javaモードとで排他的に用いているという階層的なシェアリングを示す図である。

15 図56は、第8実施形態に係るBD-ROMの製造工程を示すフローチャートである。

図57は、AVストリーム中のNavigation Button情報からダイレクトにJavaオブジェクトへと分岐するという再生制御の一例である。

発明を実施するための最良の形態

20 (第1実施形態)

以降、本発明に係る記録媒体の実施形態について説明する。先ず始めに、本発明に係る記録媒体の実施行為のうち、使用行為についての形態を説明する。図1は、本発明に係る記録媒体の、使用行為についての形態を示す図である。図1において、本発明に係る記録媒体は、BD-ROM100である。このBD-ROM100は、再生装置200、テレビ300、リモコン400により形成されるホームシアターシステムに、映画作品を供給するという用途に供される。

続いて本発明に係る記録媒体の実施行為のうち、生産行為についての形態について説明する。本発明に係る記録媒体は、BD-ROMの応用層に対する改良により実施することができる。図2は、BD-ROMの構成を示

す図である。

本図の第4段目にBD-ROMを示し、第3段目にBD-ROM上のトラックを示す。本図のトラックは、BD-ROMの内周から外周にかけて螺旋状に形成されているトラックを、横方向に引き伸ばして描画している。このト
5 ラックは、リードイン領域と、ボリューム領域と、リードアウト領域とからなる。本図のボリューム領域は、物理層、ファイルシステム層、応用層というレイヤモデルをもつ。図2に示すようなデータフォーマットを、BD-ROMの応用層上に形成することにより本発明に係る記録媒体は、工業的に生産される。

10 図3は、ディレクトリ構造を用いてBD-ROMの応用層フォーマット(アプリケーション)を表現した図である。本図に示すようにBD-ROMには、ROOTディレクトリの下にBD-ROMAVディレクトリがあり、その下にJCLASSディレクトリ、BROWSERディレクトリがある。

BD-ROMAVディレクトリの配下には、INFO.BD-ROM、XXX.M2TS、
15 XXX.CLPI, YYY.PL、ZZZ.MOVIEといったファイルが存在する。JCLASSディレクトリの配下には、ZZZ.CLASSというファイルが、BROWSERディレクトリの配下には、ZZZ.HTMLというファイルが配置されている。

図4は、機能的な観点から、これらのファイルを分類した場合の分類図である。本図において、第1層、第2層、第3層、第4層からなる階層が本図における分類を象徴的に示す。本図においてXXX.M2TSは第2層に分類される。XXX.CLPI, YYY.PLは、第3層(静的シナリオ)に分類される。BD-ROMAVディレクトリ配下のZZZ.MOVIE、JCLASSディレクトリ配下のZZZ.CLASS、BROWSERディレクトリ配下のZZZ.HTMは、第4層に分類される。

25 本図の分類(第1層～第4層)は、図5に示すようなレイヤモデルを対象とした分類である。以降、図5を参照しながら、BD-ROMが対象としている、制御ソフトウェアのレイアモデルについて説明する。

図5の第1層は、物理層であり、処理対象たるストリーム本体の供給制御である。この第1層に示すように、処理対象たるストリームは、
30 BD-ROMだけではなく、HD、メモリカード、ネットワークといったあら

ゆる記録媒体、通信媒体を供給源としている。これら HD、メモリカード、ネットワークといった供給源に対する制御(ディスクアクセス、カードアクセス、ネットワーク通信)が第1層の制御である。

第2層は、復号化方式のレイアである。第1層で供給されたストリームを、どのような復号化方式を用いて復号するのかを規定しているのがこの第2層である。本実施形態で採用する復号化方式は、MPEG2の復号化方式である。

第3層(静的シナリオ)は、ストリームの静的なシナリオを規定するレイアである。静的なシナリオとは、ディスク制作者によって予め規定された再生経路情報、ストリーム管理情報であり、これらに基づく再生制御を規定しているのがこの第3層(静的シナリオ)である。

第4層は、ストリームにおける動的なシナリオを実現するレイヤである。動的なシナリオとは、ユーザ操作や装置の状態によって再生進行を動的に変化させるためのシナリオであり、これらに基づく再生制御を規定しているのがこの第4層である。以降、このレイヤモデルに従い、ストリーム本体、静的なシナリオ、動的なシナリオにあたるファイルについて説明してゆく。

先ず第2層に属するストリーム(XXX.M2TS)について説明する。

AVストリーム(XXX.M2TS)は、MPEG-TS(Transport Stream)形式のデジタルストリームであり、ビデオストリーム、1つ以上のオーディオストリーム、1つ以上の副映像ストリーム、Navigation Button情報を多重化することで得られる。ビデオストリームは映画の動画部分を、オーディオストリームは映画の音声部分を、副映像ストリームは、映画の字幕を、Navigation Button情報は、メニューを対象とした動的な再生制御の手順をそれぞれ示している。図6は、AVストリームがどのように構成されているかを模式的に示す図である。

AVストリームは(第4段目)、複数のビデオフレーム(ピクチャpj1,2,3)からなるビデオストリーム、複数のオーディオフレームからなるオーディオストリームを(第1段目)、PESパケット列に変換し(第2段目)、更にTSパケットに変換し(第3段目)、同じくNavigation

Button 情報(第 7 段目)を、PES パケット列に変換し(第 6 段目)、更に TS パケットに変換して(第 5 段目)、これらを多重化することで構成される。この多重化は、オーディオフレームが、同じ時刻に BD-ROM から読み出されるべきビデオフレームの近くにくるように、ビデオフレーム 5 を格納した TS パケット、オーディオフレームを格納した TS パケットを並べるというものである。Navigation Button 情報は、動的な再生制御に関するものなので、ここでの説明は省略する。また副映像ストリームは、本実施形態との関連が希薄なので、図 6 での図示は省略している。

かかる過程を経て生成された AV ストリームは、通常のコンピュータ 10 ファイル同様、複数のエクステントに分割され、BD-ROM 上の領域に記録される。図 7 は AV ストリームがどのように BD-ROM に記録されるかを模式的に示す図である。

AV ストリームを構成する各エクステントが、どれだけの長さであり、BD-ROM においてどのアドレスに記録されているかは、ファイル管理情報 fk1 15 に記述される。

ファイル管理情報 fk1 には、AV ストリームを分割することで得られる 3 つのエクステント 1, 2, 3 のそれぞれについて、エクステントのアドレス(adr1, 2, 3)、長さ(length1, 2, 3)が記述されていることがわかる。 AV ストリームは、1 つ以上の ACCESS UNIT とからなり、この ACCESS 20 UNIT の単位で頭出し可能である。ACCESS UNIT とは、1 つの GOP(Group Of Picture)と、この GOP と同時に読み出されるべきオーディオフレームとを含む最小デコード単位である。GOP は、過去方向および未来方向に再生されるべき画像との時間相関特性を用いて圧縮されている Bidirectionally predictive Predictive(B)ピクチャ、過去方向に再生 25 されるべき画像との時間相関特性を用いて圧縮されている Predictive(P)ピクチャ、時間相関特性を用いず、一フレーム分の画像内での空間周波数特性を利用して圧縮されている Intra(I)ピクチャを含む。

尚、XXX.M2TS のファイル名 XXX は、BD-ROM において AV ストリームに 30 付与される 3 衔の識別番号を抽象化している。つまり本図における AV

ストリームは、この XXX を用いて一意に識別される。以上がストリーム (XXX.M2TS) についての説明である(ここでの 3 桁という桁数は例示に過ぎず、何桁でもよい。)。

5 <静的なシナリオ>

続いて、静的なシナリオであるファイル (XXX.CLPI, YYY.PL) について説明する。

ストリーム管理情報 (XXX.CLPI) は、個々の AV ストリームについての管理情報である。図 8 は、ストリーム管理情報の内部構成を示す図である。AV ストリームはビデオストリーム、オーディオストリームを多重化することで得られ、AV ストリームは ACCESS UNIT と呼ばれる単位での頭出しが可能なので、各ビデオストリーム、オーディオストリームはどのような属性をもっているか、頭出し位置が AV ストリーム内の何処に存在するかが、ストリーム管理情報の管理項目になる。図中の引き出し線はストリーム管理情報の構成をクローズアップしている。引き出し線 hn1 に示すように、ストリーム管理情報 (XXX.CLPI) は、ビデオストリーム、オーディオストリームについての「属性情報」と、ACCESS UNIT を頭出しするためのリファレンステーブルである「TMAP」とからなる。

属性情報 (Attribute) は、破線の引き出し線 hn2 に示すようにビデオストリームについての属性情報 (Video 属性情報)、属性情報数 (Number)、AV ストリームに多重化される複数オーディオストリームのそれについての属性情報 (Audio 属性情報 #1～#m) からなる。ビデオストリームについての管理情報は、破線の引き出し線 hn3 に示すようにそのビデオストリームがどのような圧縮方式で圧縮されたか (Coding)、ビデオストリームを構成する個々のピクチャデータの解像度がどれだけであるか (Resolution)、アスペクト比はどれだけであるか (Aspect)、フレームレートはどれだけであるか (Framerate) を示す。

一方、オーディオストリームについての属性情報 (Audio 属性情報 #1～#M) は、破線の引き出し線 hn4 に示すようにそのオーディオストリームがどのような圧縮方式で圧縮されたか (Coding)、そのオーディオストリ

ームのチャネル番号が何であるか(Ch.)、何という言語に対応しているか(Lang)を示す。

タイムマップ(TMAP)は、複数の頭出し位置のアドレスを、時刻情報を用いて間接参照するためのリファレンステーブルであり、破線の引き出し線 hn5 に示すように複数のエントリー情報(ACCESS UNIT#1 エントリー情報、ACCESS UNIT#2 エントリー情報、ACCESS UNIT#3 エントリー情報……)と、エントリー情報数(Number)とからなる。各エントリー情報は、引き出し線 hn6 に示すように、対応する ACCESS UNIT の再生時間(Duration)と、対応する ACCESS UNIT のデータサイズ(Size)とを対応づけてなる。可変長符号圧縮方式が採用されるため、GOP を含む各 ACCESS UNIT のサイズや再生時間がバラバラであっても、この『エントリー情報』を参照することにより、任意の再生時刻から、その再生時刻に対応する ACCESS UNIT 内のピクチャデータへと頭出しを行いうことが可能になる。尚、XXX.CLPI のファイル名 XXX は、ストリーム管理情報が対応している AV ストリームと同じ名称が使用される。つまり本図における AV ストリームのファイル名は XXX であるから、AV ストリーム(XXX.M2TS)に対応していることを意味する。以上がストリーム管理情報についての説明である。続いてプレイリスト情報について説明する。

YYY.PL(プレイリスト情報)は、再生経路情報であるプレイリストを構成するテーブルであり、CellList からなる。図 9 は、PL 情報の内部構成を示す図である。

CellList は、複数の Cell 情報(Cell 情報#1, #2, #3…#n)と、これら Cell 情報数(Number)とからなる。セル情報は、ポインタ情報であり、プレイリストを構成する 1 つ以上の論理的な再生区間を定義する。セル情報の構成は、引き出し線 hsl によりクローズアップされている。この引き出し線に示すようにセル情報は、再生区間の In 点及び Out 点が属する AV ストリームの名称を示す『Stream Name』と、再生区間の始点を示す情報『IN 点情報』と、再生区間の終点を示す情報『Out 点情報』とから構成される。

セル情報の特徴は、その表記法にある。つまりタイムマップをリファ

レンステーブルとして用いた間接参照の形式で、再生区間が定義されている。図10は、PL情報による間接参照を模式化した図である。本図においてAVストリームは、複数のACCESS UNITから構成されている。
5 ストリーム管理情報内のTMAPは、これら複数ACCESS UNITのセクタアドレスを、矢印ay1,2,3,4に示すように指定している。図中の矢印jy1,2,3,4は、CELL情報によるACCESS UNITの参照を模式化して示している。つまり、CELL情報による参照(矢印jy1,2,3,4)は、TMAPを介すことにより、AVストリーム内に含まれる複数ACCESS UNITのアドレスを指定するという間接参照であることがわかる。

10 Cell情報—ストリーム管理情報—AVストリームの組みからなるBD-ROM上の再生区間を『セル』という。PL情報—ストリーム管理情報—AVストリームの組みからなるBD-ROM上の論理的な再生単位を『プレイリスト(PLと略す)』という。BD-ROMに記録された映画作品は、この論理的な再生単位(PL)にて構成される。論理的な再生単位にて、BD-ROMにおける映画作品は構成されるので、本編たる映画作品とは別に、あるキャラクタが登場するようなシーンのみを指定するようなPLを定義すれば、そのキャラクタが登場するシーンのみからなる映画作品を簡単に制作することができる。図11は、図10に示したPL情報(PL情報#1)とは、別のPL(PL情報#2)を定義する場合の一例を示す図である。

15 20 様々なPL情報を定義するだけで、映画作品のバリエーションは増えるので、映画制作者の表現の幅を増やすことが、静的なシナリオの最大のメリットである。

また、BD-ROMにおける再生単位には、PL、CELLといったものその他、Chapterがある。Chapterは、1つ、2つ以上のCELLから構成される。

25 尚、PL情報のファイル名YYYは、BD-ROMにおいてPL情報に付与される3桁の識別番号を抽象化している。つまり本図におけるPL情報は、この識別番号YYYを用いて一意に識別される。PL情報の識別番号を”YYY”と表現しているのは、PL情報の識別番号が、AVストリーム及びAVストリーム管理情報の識別番号XXXとは別の番号体系であることを30意味している(ここでの3桁という桁数は例示に過ぎず、何桁でもよ

い。)。

以上が、静的なシナリオについての説明である。続いて動的なシナリオについて説明する。

5 <動的なシナリオ>

動的なシナリオは、AVストリームの再生制御を示すプログラムである。動的なシナリオによる再生制御は、装置に対するユーザ操作に応じて変化するものであり、プログラム的な性質をもつ。ここでの動的な再生制御には、2つのモードがある。2つのモードのうち1つは、AV機器特有の再生環境で、BD-ROMに記録された動画データを再生するモード(ノーマルモード)であり、もう1つはBD-ROMに記録された動画データの付加価値を高めるモード(エンハンスドモード)である。図12は、レイヤモデルの第4層における再生モードを示す図である。本図において第4層には、1つのノーマルモードと、2つのエンハンスドモードとが記述されている。1つのノーマルモードは、DVDライクな再生環境での再生モードでありMOVIEモードと呼ばれる。2つのエンハンスドモードのうち、1つ目は、Java仮想マシンを主体とした再生モードであり、Javaモードと呼ばれる。2つ目のエンハンスドモードのうち、2つ目はブラウザを主体とした再生モードであり、Browserモードと呼ばれる。

第4層には、MOVIEモード、Javaモード、Browserモードという3つのモードがあるので、動的な再生制御はどれかのモードで実行できるよう記述されればよい。DVDプレーヤ向けのコマンドと良く似たコマンドで、制御手順を記述したい場合は、既存のDVD再生装置と良く似た再生制御を再生装置に実行させることができる。ページ記述言語で制御手順を記述した場合は、ネットワーク上のサイトをアクセスしたり、ファイルをダウンロードするような制御手順を記述することができる。図13は、3つのモードの動的な再生制御にて作成される映画作品を示す図である。

図13(a)は、MOVIEモードで動的な再生制御を定義することにより、作成される映画作品の一場面を示す図である。MOVIEモードはDVD

再生装置が解釈可能なコマンドと良く似たコマンドで再生制御を記述することができるので、DVDと同じような再生制御、つまり、メニューに対する選択により再生が進行するような再生制御を定義することができる。

5 図13(b)は、Javaモードで動的な再生制御を定義することにより、作成される映画作品である。JavaモードはJava仮想マシンが解釈可能なJava言語で制御手順を記述することができる。この再生制御がコンピュータ・グラフィックス(CG)の動作を制御するものなら、Javaモードにあっては、動画を表示した画面の横でCG(図中のフクロウの10 絵)が動きまわっているような再生制御を定義することができる。

図13(c)は、Browserモードで動的な再生制御を定義することにより、作成される映画作品である。Browserモードはブラウザが解釈可能なHTML言語で再生制御を記述することができる。この再生制御がWEBサイトのアクセスやウィンドウの動作を制御するものなら、15 Browserモードにあっては、動画を表示した画面の横で、WEBサイトから得たオンラインデータ(図中のキャラクタAの掲示板やバナー広告)を表示するような再生制御を定義することができる。

以上が動的なシナリオの概略である。以降、各のモードの再生制御を規定するファイルについて、モード毎に説明してゆく。

20

<MOVIEモードにおける動的なシナリオ>

先ずMOVIEモードにおける動的シナリオについて説明する。MOVIEモードにおける動的シナリオには、トランSPORTストリーム中のNavigation Button情報、MOVIEオブジェクトがある。

25 Navigation Button情報とは、AVストリーム内に多重化されたストリームの1つであり、メニューにおけるボタンの挙動を司り、確定されたボタンに応じた再生制御を行う。メニューの挙動には、リモコンの矢印キーの押下に応じてメニュー上のボタンの状態を変えるというもの(1)、メニュー上のボタン確定に応じて再生装置におけるレジスタの値を更新30 するというもの(2)、メニュー上のボタン確定に応じて分岐を実現する

というもの(3)があり、これらの挙動を制御し、ボタンに応じたコマンドを再生装置に実行させることが、Navigation Button 情報の役割である。図14は Navigation Button 情報の構成を示す図である。

Navigation Button 情報は、ボタン制御情報(Button Control 情報)と、
5 イメージデータ(Image Data)とからなる。イメージデータは、破線の引き出し線 hh1 に示すように複数の PNG データ(PNGs)と、これら PNG データが共通に参照する Look Up Table(COMMON LUT)とからなる。イメージデータにおける個々の PNG データ(PNG データ#1, #2, #3)は、メニューにおける個々のボタン(Button#1, #2, #3)を描いたイメージデータである。

10 ボタン制御情報は、破線の引き出し線 hh2 に示すように Page Affiliation 情報と Button Affiliated 情報とからなる。Page Affiliation 情報は、破線の引き出し線 hh3 に示すようにメニューにおけるボタン表示開始時刻を示す『Button Display Begin Time』、表示終了時刻を示す『Button Display End Time』、初期状態において選択状態にすべきボタンを示す『Initially Selected Button』とからなる。
15

Button Affiliated 情報は、破線の引き出し線 hh4 に示すように複数の Button Affiliation#1, #2……からなる。各 Button Affiliation は、破線の引き出し線 hh5 に示すように複数の PNG データのうち、ボタンのイメージがどれであるかを示す『Button Image 情報』、ボタンイメージの配置位置を示す『Button display Location』、各ボタンの上下左右にどのボタンが存在するかを示す『Relationship with Upper/Lower/Side Buttons』、各ボタンの確定時に実行すべき『Button Command』からなる。

20 メニューにおける各ボタンの Button Affiliated 情報には、ボタンの上下左右にどのボタンが存在するかが記されているので、ユーザによりリモコンの矢印キーが押下されれば、Button Affiliated 情報における Upper/Lower/Side Buttons を参照して、その矢印キーの方向に沿った位置にあるボタンを知得し、そのボタンの状態を変化させることができる。これにより、矢印キーの押下に応じて、そのキーの方向のボタンの状態が変化することになる。そしてユーザにより確定操作がなされた場
25
30

合、そのボタンの Button Affiliated 情報における Button Command を実行することにより、ボタンの押下に応じた動的な再生制御を実行することができる。

Navigation Button 情報は AV ストリームに組み込まれているので、
5 ある動画の一コマが画面に現れたタイミングに、特定の処理を再生装置
に実行させるという再生制御、つまり、動画内容と緻密に同期した再生
制御の記述に便利である。また Navigation Button 情報は、AV ストリ
ーム自身に多重化されているので、再生制御を行いたい区間が数百個で
あっても、それらに対応する Navigation Button 情報の全てをメモリに
10 格納しておく必要はない。Navigation Button 情報は ACCESS UNIT 毎に
ビデオパケットと共に BD-ROM から読み出されるので、現在再生すべき
動画区間に対応する Navigation Button 情報をメモリに常駐させ、この
動画区間の再生が終われば、その Navigation Button 情報をメモリから
削除して、次の動画区間に対応する Navigation Button 情報をメモリに
15 格納すればよい。Navigation Button 情報は、AV ストリームに多重化さ
れるので、たとえ Navigation Button 情報の数が数百個になってもメモ
リの搭載量を必要最低限にすることができる。

Navigation Button 情報におけるコマンドの記述例を図 15 を参照し
ながら説明する。図 15 は、AV ストリームに多重化されている
20 Navigation Button 情報でボタンの制御を行う場合の一例を示す図であ
る。

図 15 における AV ストリームは、映画であり、図中の複数ボタンは
この映画の一コマに合成されて表示される。図中のボタン A, B, C は映画
作品に登場する各キャラクタに対応するものであり、この映画に登場す
25 るキャラクタ A, B, C の何れかを選ぶという選択操作により選択状態、確
定状態と状態を変える。

本図において Navigation Button 情報は、ビデオストリームにおいて
対話操作が必要な期間より前に配置される。図中の AV ストリームに多
重化されている Navigation Button 情報は、メニューにおける各ボタン
30 の確定時に、それぞれ異なる値を GPRM(0)に設定する。GPRM(0)とは、

レイアモデルの第3層(静的シナリオ)(静的なシナリオ)で管理されるレジスタ設定値である。具体的にいうと、キャラクタAのボタンの確定時には、GPRM(0)が1に設定され、キャラクタBのボタンの確定時には、GPRM(0)が2に設定される。キャラクタCのボタンの確定時には、
5 GPRM(0)が3に設定される。このようにNavigation Button情報を描画すれば、メニューの描画時にどのボタンが選択されたかという情報をGPRM(0)に残しておくことができる。以上がNavigation Button情報についての説明である。

MOVIEオブジェクト(XXX.MOVIE)とは、DVD再生装置が解釈可能なコマンドに類似したコマンドにて記述された、動的なシナリオである。PLの再生を命じる再生コマンド、その再生に先立ち実行すべきコマンド(PREコマンド)、その再生後に実行すべきコマンド(POSTコマンド)からなる。1つ以上のMOVIEオブジェクトと、各MOVIEオブジェクトにて再生が命じられるPLとの組みをTitleという。Titleは、BD-ROMにおける映画作品全体に相当する単位である。またMOVIEオブジェクトは、M-
10 OBJと略記する場合がある。
15

図16は、AVストリーム中のNavigation Button情報を図15のように設定されている場合のMOVIEオブジェクトの一例を示す。本図のMOVIEオブジェクトは、GPRM(0)に0を設定するPREコマンド、PLの再生を再生装置に命じるコマンド(PlayPl(PL#1))、他の動的シナリオへの分岐を再生装置に命じるPOSTコマンド(IF(GPRM(0)=0){Jump
20 Title#m} else{Jump Title#m+1})からなる。このPREコマンドにより、PLの再生に先立ち、GPRM(0)が初期化されることになる。またPOSTコマンドによりGPRM(0)が初期化時の0を示すなら、MOVIEオブジェクト#m+1に分岐することになる。一方、メニュー表示時にボタン選択がな
25 され、GPRM(0)が0以外の値に設定されたなら、他のタイトル>Title#m)に分岐することになる。

ユーザ操作に応じてレジスタの設定値を変えるコマンドを
Navigation Button情報を組み込んでおき、再生装置におけるレジスタ
30 設定値を用いた条件付き分岐をPOSTコマンドとして記述しておけば、

ユーザ操作に応じて再生進行が変わらるような映画作品(マルチストーリー型の映画)を、容易に作成することができる。

MOVIE モードにおけるシナリオには、Navigation Button 情報、MOVIE オブジェクトの 2 種があるので、メニューにおけるボタンの挙動と緻密 5 に同期させたい処理は Navigation Button 情報に記述し、PL 再生の前 処理、後処理というような統合的な処理については、MOVIE オブジェクトに記述することができる。ボタンと同期させるべき処理か、統合的な 処理かによって、コマンドの記述を変えることができるので、再生制御 の表現の幅が広がる。またボタンと同期するような再生制御の数を増や 10 しても、古い Navigation Button 情報を新たな Navigation Button 情報 で上書きしてゆくことで、かかる再生制御の記述には再生装置のメモリ の大規模化を招かない。

<Java モードにおける動的なシナリオ>

続いて Java モードで用いられる動的シナリオについて説明する。

15 ZZZ.CLASS は、Java モードにおける動的な再生制御を規定するアプリケーションプログラム(Java オブジェクト)を定義したクラスファイル である。Java モードでは、Java 仮想マシンがシナリオの実行主体となるので、Java オブジェクトは Java 言語で記述される。Java 言語(登録 商標)は SUN Microsystems 社が開発したミドルウェア向けの記述言語 20 であり、近年では日本の携帯電話や欧州のディジタル放送 DVB-MHP など 民生機器で広く使われつつある。Java 言語は C++などと同じオブジェクト 指向のプログラミング言語である。C++との違いは、C++はオペレーティングシステム上で実装されているのに対して、Java は仮想マシーン(Java Virtual Machine)を定義し、Windows や Linux などの主要オペレーティングシステム上で実装されている。そのため Java 言語を用いれば、オペレーティングシステムに依存することのない処理手順を記述する 25 ことができる。Java 言語が携帯電話や STB に採用されているのは、 実行環境がメーカー毎に異なっていても、実行環境に依存しない処理手順を記述することができるからである。Java オブジェクトは、J-OBJ と略 30 記する場合がある。

図17は、図16のMOVIEオブジェクトのコマンドにより分岐がなされた場合のJavaオブジェクトの記述例である。このJavaオブジェクトは、CGを描画し(1)、PL中のCELLを再生するという処理(2)を、GPRM(0)の値に応じて行うものである。こうした処理により、図13
5 (b)に示すような合成画像、つまり、PLを構成する動画データを再生させながら、その脇で、キャラクタのCGを動作させるという合成画像を作成することができる。またこの記述例のif文は、GPRM(0)に応じて、描画すべきキャラクタや、再生すべきCELLを変化させている。

描画すべきCGに着目して説明を行う。図17の一例において、
10 『A.DrawCharacter();』とは、キャラクタAというClassの1つのmethod(※図中のDrawCharacter関数のことである)を用いて、キャラクタAのObjectを画面上に描画されることを意味する。

『B.DrawCharacter();』、『C.DrawCharacter();』も同じ意味であり、
15 キャラクタB,CというClassの1つのmethod(※図中のDrawCharacter関数のことである)を用いて、キャラクタB,CのObjectを画面上に描画されることを意味する。

『A.DrawCharacter();』、『B.DrawCharacter();』、
『C.DrawCharacter();』は、GPRM(0)の値によって排他的に実行される
ので(図中のif文)、GPRM(0)が1ならキャラクタAのCGが、GPRM(0)が
20 2ならキャラクタBのCGが、GPRM(0)が3ならキャラクタCのCGが、
それぞれ描画される。図18は、MOVIEオブジェクトとJavaオブジェクトとの関連を示す図である。図18の左半分は、図16に示した
MOVIEオブジェクトの記述例と同一である。本図においてGPRM(0)は、
AVストリームに組み込まれたNavigation Button情報により設定され
25 る。GPRM(0)は、レイアモデルにおいて第3層(静的シナリオ)(静的シナリオ)にて管理されるパラメータである。このGPRM(0)は、MOVIEモード、
Javaモード、Browserモードの何れからでも参照されるので、Java
オブジェクトにおける処理を、MOVIEモードにおいて設定された
GPRM(0)に応じて切り換えることができる。MOVIEモードにおける
30 Navigation Button情報により、GPRM(0)の値を変化させ、Javaモード

において、描画すべき CG を切り換えることができる。GPRM(0)による MOVIE オブジェクトと Java オブジェクトとの連携により、動画データの再生と、Java 仮想マシンによる処理とを融合させた新しいスタイルの映画作品を創作することができる。

5 GPRM(0)の値に応じた Java オブジェクトの処理が、キャラクタ A, B, C のどれかを主役にしたゲームであるなら、MOVIE モードにおいて動画でキャラクタの紹介を行い、ユーザによるキャラクタ選択を受け付けて、Java モードにおいてユーザが選択したキャラクタを主役にしたゲームを実行するという手順(このような手順は現在のゲームソフトに多くみられる)を、映画の動画データを用いて簡単に記述することができる。

10 図 19 は、MOVIE オブジェクトにおけるキャラクタ紹介と、Java モードにおけるゲームとから構成されるタイトルを模式的に示す図である。

GPRM(0)を介した MOVIE モードと Java モードとの連携により、キャラクタの紹介から、そのキャラクタを主役にしたゲーム実行までの一連の手順を、MOVIE モードにおける映画を用いて簡単に記述することができる。

15 一方、再生すべき CELL について着眼すると、GPRM(0)が 1 なら PL#1 の CELL#1 が、GPRM(0)が 2 なら PL#2 の CELL#1 が、GPRM(0)が 3 ならキャラクタ C が、それぞれ描画される。

ここで、PL#1 の CELL#2 は、AV ストリームのうち、キャラクタ A が登

20 場している区間のみを指定している再生単位であるものとする。この CELL を、キャラクタ A の CG と共に再生させることで、キャラクタ A の

CG が動きまわり、その横で動画像たるキャラクタ A が表示されるよう

な合成画面(図 13 (b))を表示させることができる。図 20 は、Java オブジェクトからの PL 再生を模式的に示した図である。本図における

25 Java オブジェクトは、PL#1 の CELL#2 の再生を命じる関数呼出を含む。

この関数呼出により、PL を構成する 1 つの CELL が再生されることにな

る。本図における PL#1 は、MOVIE モードにおいて再生が命じられているものであれば、この PL#1 のうち、CELL#2 の部分区間のみが、Java オ

30 オブジェクトにおいて引用されることになる。このような Java オブジェクトからの関数呼出により、MOVIE モードにおいて再生されるべき映像

作品(PL#1)のうち、特定のシーン(CELL#2)のみを Java オブジェクトにおけるゲームで用いるという”シーンの引用”が容易に行われる。図 20 の一例は、MOVIE モードで使用された PL を Java モードで利用するという、PL のシェアリングを前提としたシーン引用である。PL をシェアリングするのではなく、MOVIE モードで使用されたいた PL とは別の PL を用いて、シーン引用することも可能である。図 21 は、MOVIE モードで使用されたいた PL とは別の PL を用いることによる、シーン引用を示す図である。

これら図 20、図 21 の引用により、映画の中のシーンを Java モードにおけるゲームの中で回想するという回想シーンを容易に作成することができる。かかる引用により、映画作品制作にあたっての表現の幅が広がり、BD-ROM の商品価値を高めることができる。尚、図 21、図 22 の一例は、CELL の単位で引用を行ったが、Java オブジェクトから再生を命じる範囲を、時間で指定してもよい。同様に Browser モードから引用を行ってもよい。映画作品との連携を図ったゲームソフトや、映画作品の一シーンを引用したゲームソフトを Java 言語にてプログラミングさせてるので、BD-ROM の制作事業に、多くのソフトハウスを参入させることができる。

20 <Browser モードにおける動的なシナリオ>

WebPage オブジェクト(ZZZ.HTM)とは、HTML, XML, BML 等のページ記述言語を用いて記述されたシナリオである。WebPage オブジェクトは、WP-OBJ と略記する場合がある。HTML での記述が可能なので、Browser モードは WEB サイトへのアクセスを交えた制御手順を記述することができる。図 22 は、WebPage オブジェクトの記述例である。この記述例は、図 18 をベースにしている。本図の WebPage オブジェクトが異なるのは、GPRM(0)の値に応じて異なる WEB サイトにアクセスしている点である。つまり、GPRM(0)が 1 である場合、キャラクタ A に関連する WEB サイトを、GPRM(0)が 2 である場合、キャラクタ B に関連する WEB サイトを、GPRM(0)が 3 である場合、キャラクタ C に関連する WEB サイトを、それ

ぞれアクセスしている。本図において GPRM(0)は、AV ストリームに組み込まれた Navigation Button 情報により設定される。また GPRM(0)は、レイアモデルにおいて第 3 層(静的シナリオ)に存在しており、MOVIE モード、Java モード、Browser モードの何れからでも参照されるので、5 WebPage オブジェクトにおける処理を、MOVIE モードにおいて設定された GPRM(0)に応じて切り換えることができる。WEB サイトからのオンラインデータが、毎日、毎週といった間隔で更新されるニュースや BBS、バナー広告である場合、これらのオンラインデータの表示を交えることで、動画データに対する印象を常に新鮮なものに保つことができる。また 10 WebPage オブジェクトでは、WEB サーバにアクセスする手順の記述が可能なので、最新バージョンの PL や VOB を WEB サーバから取得し、これらを交えた再生制御を再生装置に行わさせることができる。GPRM(0)による、MOVIE オブジェクトと WebPage オブジェクトとの連携により、動画データの再生と、ブラウザによる処理とを融合させた新しいタイプ 15 の映像作品を創作することができる。

尚、ファイル名 ZZZ.MOVIE、ZZZ.CLASS、ZZZ.HTM におけるファイルボディ「ZZZ」は、BD-ROM において動的シナリオに付与される 3 衔の識別番号を抽象化している。つまり本図におけるシナリオは、この識別番号 ZZZ を用いて一意に識別される。シナリオの識別番号を”ZZZ”と表現しているのは、シナリオの識別番号が、AV ストリームの識別番号 XXX、PL 情報の識別番号 YYY とは別の番号体系であることを意味している(ここでの 3 衔という桁数は例示に過ぎず、何桁でもよい。)。

<シナリオの記述手法>

25 MOVIE オブジェクト、Java オブジェクト、WebPage オブジェクトの制御手順を記述するための記述手法について説明する。これらのシナリオが対象としているレイアモデルにおいて、MOVIE オブジェクトを実行する DVD 仮想プレーヤ、Java オブジェクトを実行する Java 仮想マシン、WebPage オブジェクトを実行するブラウザは、何れも第 3 層(静的シナリオ)上に存在している。Java オブジェクトや WebPage オブジェクトは、
30

CG の描画や WEB サイトのアクセス等、Java モードやブラウザに行わせるべき本来の処理を、Java 言語や HTML で記述し、これ以外の処理、つまり BD-ROM の再生制御は、第 3 層(静的シナリオ)から供給されるプログラミング関数を用いて記述すればよい。

5 以降、第 3 層(静的シナリオ)から供給される関数について説明する。

(a) 再生関数：第 1 引数で指定するプレイリストの再生を、第 2 引数で指定する位置から再生を開始させる；

10 書式: PlayPL(第 1 引数, 第 2 引数)

第 1 引数は、プレイリストの番号で、再生すべき PL を指定することができる。第 2 引数は、その PL に含まれる CELL や、その PL における任意の時刻、Chapter、Mark を用いて再生開始位置を指定することができる。

15 CELL により再生開始位置を指定した PlayPL 関数を PlayPLatCELL()、
Chapter により再生開始位置を指定した PlayPL 関数を
PlayPLatChapter()、

Mark により再生開始位置を指定した PlayPL 関数を PlayPLatMark()、
時刻情報により再生開始位置を指定した PlayPL 関数を
20 PlayPLatSpecified Time() という。

(b) 再生装置の状態取得や状態設定のための関数

再生装置の状態は、32 個の Player Status Register(この設定値は、
System Parameter(SPRM)と呼ばれる)と、32 個の General Purpose
25 Register(この設定値は、General Parameter(GPRM)と呼ばれる)とに
示されている。

MOVIE オブジェクト、Java オブジェクト、WebResponse オブジェクトは、
以下の(i)～(iv)の関数を使用することにより、これらのレジスタに値
を設定したり、これらのレジスタから値を取得したりすることができる。

(i) Get value of Player Status Register 関数

書式：Get value of Player Status Register(引数)

この関数は、引数で指定された Player Status Register の設定値を取得する関数である。

5

(ii) Set value of Player Status Register 関数

書式：Set value of Player Status Register(第1引数、第2引数)

この関数は、第1引数で指定された Player Status Register に、第2引数で指定された値を設定させる関数である。

10

(iii) Get value of General Purpose Register 関数

書式：Get value of General Purpose Register(引数)

この関数は、引数で指定された General Purpose Register の設定値を取得する関数である。

15

(iv) Set value of General Purpose Register 関数

書式：Set value of General Purpose Register(第1引数、第2引数)

この関数は、第1引数で指定された General Purpose Register に、第2引数で指定された値を設定させる関数である。

Player Status Register の設定値(SPRM)がどのような意味をもつかは、以下に示す通りである。以下の SPRM(x)という表記は、x 番目の Player Status Register の設定値を意味する。

25

SPRM(0) : Reserved

SPRM(1) : デコード対象たるオーディオストリーム
のストリーム番号

SPRM(2) : デコード対象たる副映像ストリームのストリー
ム番号

30

	SPRM(3)	: ユーザによるアングル設定を示す番号
	SPRM(4)	: 現在再生対象とされているタイトルの番号
	SPRM(5)	: 現在再生対象とされている Chapter の番号
	SPRM(6)	: 現在再生対象とされている PL の番号
5	SPRM(7)	: 現在再生対象とされている CELL の番号
	SPRM(8)	: 現在の再生時点を示す時刻情報
	SPRM(9)	: ナビゲーションタイマのカウント値
	SPRM(10)	: 現在選択状態にあるボタンの番号
	SPRM(11)～(12)	: Reserved
10	SPRM(13)	: ユーザによるパレンタルレベルの設定
	SPRM(14)	: 再生装置の映像再生に関する設定
	SPRM(15)	: 再生装置の音声再生に関する設定
	SPRM(16)	: 再生装置における音声設定を示す言語コード
	SPRM(17)	: 再生装置における字幕設定を示す言語コード
15	SPRM(18)	: メニュー描画のための言語設定
	SPRM(19)～(31)	: Reserved

{SPRM(1)～(2)について}

SPRM(1)～(2)は、PL 再生に先立ち PRE コマンドにより設定される。

20 また AV ストリームの再生中において、Navigation Button 情報における Button Command によって更新される。SPRM(1)(2)を参照することで、以下のような応用が可能である。例えば、MOVIE モードにおいて音声を英語に設定し、字幕を日本語に設定して、PL の再生を行っているものとする。この再生時に、MOVIE モードから Java モードへの分岐があれば、Java オブジェクトは SPM(1), (2)を参照することにより、音声を英語に設定し、字幕を日本語に設定した上で Java オブジェクトのソフトウェアを実行することができる。Java オブジェクトのソフトウェアが英語リスニングの実習教材であるなら、映画の視聴と、リスニング教材による実習との相乗効果により、語学の習得効果を高めることができる。

30 {SPRM(3)について}

SPRM(3)は、PL 再生に先立ち PRE コマンドにより設定される。また AV ストリームの再生中において、Navigation Button 情報における Button Command によって更新される。AV ストリームにマルチアングル区間が存在する場合、どのアングルの動画データがデコード対象であるかは、
5 SPRM(3)を参照すればわかる。

SPRM(3)を参照することで、以下の応用が可能である。映画作品が列車車窓ビデオであり、マルチアングル区間を含んでいるものとする。このマルチアングル区間は、右側座席、左側座席、運転席といった複数アングルから撮影された複数動画データを含んでいるものとする。この場合、SPRM(3)には、どのアングルからの再生かが示されているので、
10 MOVIE モードから Java モードへの分岐時において、Java オブジェクトが SPrM(3)を参照すれば、ユーザ設定されたアングルからの映像再生を伴う運転ゲームを実現することができる。

{SPRM(4)～(7)について}

SPRM(4)は、メニュー操作を介して、ユーザにより Title が選択された場合に更新される。SPRM(5)～(7)は、現在の再生時点が進行する度に更新される。つまり現在の再生時点がある CELL から別の CELL に移行すれば、SPRM(7)が更新される。ある PL から別の PL への切り換えがなされれば SPrM(6)が更新される。Chapter の切り換えがなされれば
20 SPrM(5)が更新される。

これにより再生装置が、どの Title を再生しているか、どの PL を再生しているか、またその PL においてどの CELL、どの Chapter を再生しているか、どの CELL を再生しているのかは、SPRM(4)～(7)を参照すればわかる。尚 SPrM(4)～(7)は、必ずしも PRE コマンド、POST コマンド、
25 Button コマンドによりダイレクトに更新されるものではないが、PL 再生というコマンドにより更新を受ける。間接的であるにせよ、動的シナリオから更新されることには変わりないので、SPRM(4)～(7)は動的シナリオにより更新されるといえる。

30 {SPRM(10)について}

SPRM(10)は、AV ストリームに属する各ピクチャデータが表示される度に更新される。つまり再生装置が新たなピクチャデータを表示させれば、その新たなピクチャデータの表示開始時刻(Presentation Time)を示す値に SPrM(10)は更新される。この SPrM(10)を参照すれば、PL 再生の途中で MOVIE モードから Java モードへと移行するように Navigation Button 情報内の Button Command が記述されている場合、SPrM(10)を参照することで、Java モードにおける Java オブジェクトは、ユーザが同じ BD-ROM に記録されている映画作品をどこまで視聴したかという視聴位置を知得することができる。そして Java モードでは、この視聴位置に応じて登場するキャラクタが変わるように、ゲームソフトを記述することができ、より Java モードにおけるゲームをより面白くすることができます。尚 SPrM(10)は、PRE コマンド、POST コマンド、Button コマンドによりダイレクトに更新されるものではないが、PL 再生というコマンドにより更新を受ける。間接的であるにせよ、動的シナリオから更新されることには変わりないので、SPrM(10)は動的シナリオにより更新されるといえる。

Get value of Player Status Register 関数、Get value of General Purpose Register 関数を用いて以上のような Player Status Register を参照すれば、Java オブジェクト及び WebPage オブジェクトは再生装置の状態を詳細に知得することができる。

(C)第 3 層(静的シナリオ)から供給されるプログラミング関数ではないが、動的シナリオから動的シナリオへの分岐も存在する。動的シナリオから動的シナリオへの分岐を行わせる関数には、以下の JMP 関数と、
CALL 関数がある。

JMP 関数

書式：JMP 引数

CALL 関数

書式：CALL 引数

JMP 関数は、現在の動的シナリオを途中で廃棄し(discard)、引数たる分岐先動的シナリオを実行するという分岐である。JMP 命令の形式には、分岐先動的シナリオを直接指定している直接参照のものと、分岐先動的シナリオを間接参照している間接参照のものがある。

5 Call 関数とは、動的シナリオの動作を一旦停止してから(Suspend)、引数たる分岐先動的シナリオを動作させ、分岐先動的シナリオの動作終了後、停止中の動的シナリオの動作を再開するという分岐である。Call 命令の分岐先となる動的シナリオは、その末尾に Resume 命令が配置されている。Resume 命令とは、いわゆるサブルーチンの Return 命令であり、Call 関数を実行したために Suspend 状態になっている動的シナリオを、再起動させるための命令である。Call 関数の形式には、JMP 命令同様、分岐先動的シナリオを直接指定している直接参照のものと、分岐先動的シナリオを間接参照している間接参照のものがある。
10

以上が、第 3 層(静的シナリオ)により供給される関数及び変数である。

15

<統合管理の情報>

続いて、MOVIE モードにおける動的シナリオ、Java モードにおける動的シナリオ、Browser モードにおける動的シナリオを統合管理するための情報について説明する。かかる統合管理のための情報は、図 3 に示した INFO.BD-ROM である。
20

図 23 (a) は、INFO.BD-ROM の内部構成を示す図である。本図に示すように INFO.BD-ROM は Index Table を含む。Index Table は、動的シナリオから動的シナリオへの分岐の際、参照される間接参照用テーブルであり、複数ラベルのそれぞれに対する Index からなる。各 Index には、
25 そのラベルに対応する動的シナリオのファイル名が記述されている。図
23 (b) に示すように、このファイル名は、ファイルボディと、拡張子とからなる。ラベルには、Title#1～#m、Title#m+1～#n、Title#0 といったものがある。Index Table は、3 つのモードの何れの動的シナリオからも参照される。MOVIE オブジェクトから Java オブジェクト、
30 MOVIE オブジェクトから WebPage オブジェクトへの分岐は、Index

Table を介した場合のみ可能となる。逆に言えば、Index Table に Index がない Java オブジェクト、WebPage オブジェクトには、MOVIE オブジェクトからは分岐できない。

Title#1～#mINDEX は、BD-ROM において 1 から m 番目にエントリーされている Title についての Index であり、これら 1 から m までの Title 番号の選択時において分岐先となる MOVIE オブジェクトのファイル名が記述される。図 23 (b) は、Title#1～#mINDEX の内容を示す。本図に示すように Title#1～#mINDEX には、MOVIE オブジェクトのファイル名が記述されている。このファイル名は、ファイルボディ (ZZZ)、拡張子 (.MOVIE) からなる。

Title#m+1～#nINDEX は、BD-ROM において m+1 から n 番目にエントリーされている Title についての Index であり、これら m+1 から n までの Title 番号の選択時において分岐先となる WebPage オブジェクト／Java オブジェクトのファイル名が記述される。図 23 (c) は、Title#m+1～#nINDEX の内部を示す。本図に示すように Title#m+1～#nINDEX には、Java オブジェクトのファイルボディ (ZZZ)、拡張子 (.MOVIE) か、又は、WebPage オブジェクトのファイルボディ (ZZZ)、拡張子 (.HTM) が格納されている。

Title#0INDEX は、エンハンスドモードが実行不能な再生装置に装填され、エンハンスドモードへの分岐が再生装置に命じられた場合に参照された場合、エンハンスドモードシナリオの代わりに実行すべき、ムービーモードシナリオのファイル名を格納しておく Index である。

エンハンスドモードの実行不能とは、Java モードを実行する Java 仮想マシンや、Browser モードを実行するブラウザがインストールされていないか(1)、アンインストールされているか(2)、またインストールされてはいるが、ネットワークからは遮断され、スタンドアローンで用いられている(3)ことを意味する。以上の(1)～(3)の理由で、エンハンスドモードの実行が不可能な状態の再生装置をコアシステムという。また Java 仮想マシン、ブラウザによるプログラムの実行が可能な状態にある再生装置をフルシステムという。以下、コアシステム、フルシステム

による BD-ROM の間接参照について、図 24 を参照しながら説明する。

この間接参照の説明では、図 24 (a) に示すような複数の動的シナリオ (001.MOVIE, 002.MOVIE, 003.MOVIE……、

01.CLASS, 002.CLASS, 003.CLASS……) が記録されている BD-ROM を想定

5 している。BD-ROM に図 24 (a) に示すような複数動的シナリオが記述されている場合の Index Table の記述例を図 24 (b) に示す。図 2

4 (b) の記述例において、Title#1Index～Title#mIndex には、ムー

ビーモードシナリオである (001.MOVIE, 002.MOVIE, 003.MOVIE……) が記述されている。一方、Title#m+1Index～Title#nIndex には、エンハン

10 スドモードシナリオのファイル名

(001.CLASS, 002.CLASS, 003.CLASS……) が記述されている。図 25

(a) は、図 24 (b) のように Index Table が記述された場合のフル

システムにおける間接参照を示す。Index Table がこのように記述され

ているので、ラベル Title#1～Title#m を分岐先とした分岐コマンドの

15 実行時には、ラベル Title#1～Title#m を分岐先とした分岐コマンドの

実行時には、Title#1Index～Title#mIndex からファイル名

『001.MOVIE, 002.MOVIE, 003.MOVIE……』が、ラベル Title#m+1～

Title#n を分岐とした分岐コマンドの実行時には、Title#m+1Index～

Title#nIndex からファイル名『001.CLASS, 002.CLASS, 003.CLASS…』

20 が取り出され、そのファイル名の動的シナリオがメモリに読み出されて、

実行されることになる。これがフルシステムによる間接参照である。

図 25 (b) は、コアシステムにおける間接参照を示す図である。ラ

ベル Title#1～Title#m を分岐先に指定した分岐コマンドの実行時には、

図 25 (a) と同様、Title#1Index～Title#mIndex からファイル名

25 『001.MOVIE, 002.MOVIE, 003.MOVIE……』が取り出されるが、ラベル

Title#m+1～Title#n を分岐先に指定した分岐コマンドの実行時には、

Title#m+1Index～Title#nIndex に代えて Title#0Index からファイル名

『000.MOVIE』が取り出されて、そのファイル名の動的シナリオが再生

装置により実行される。以上がフルシステム、コアシステムにおける間

30 接参照である。

図26は、図18に示したMOVIEオブジェクトからJavaオブジェクトへの分岐がどのように行われるかを模式的に示す図である。図中の矢印jn1,2は、MOVIEオブジェクトからJavaオブジェクトへの分岐を象徴的に示している。本図におけるJmp Title#m+1がJavaオブジェクトにおける分岐コマンドであり、ラベルTitle#m+1のIndexを介した間接参照の形式で分岐先のJavaオブジェクトを指定している。ラベルTitle#m+1のIndexにはJavaオブジェクトのファイル名が記述されており、これを参照することにより、再生装置はどのファイルをJavaオブジェクトとして読み出せばよいかを知ることができる。

図27は、コアシステムの再生装置に図18に示したシナリオが記録されたBD-ROMが装填された場合、どのような分岐が行われるかを示す。図18に示した矢印が、図27において破線hs1で描かれているのは、コアシステムでは、Javaオブジェクトを実行する主体が存在しないので、図18に示した分岐が無効化されていることを示す。図中の実線矢印js1は、この分岐に代えてなされる予備の分岐を示す。この予備の分岐は、Title#0のIndexを介した間接参照である。Title#0のIndexには、MOVIEオブジェクト0のファイル名が格納されており、この分岐ではMOVIEオブジェクト0が再生装置により読み出され、実行されることになる。sg1は、MOVIEオブジェクト0の分岐時に実行されるMOVIEオブジェクトである。コアシステムのみの再生装置に本BD-ROMが装填された場合、MOVIEモードのゲームをユーザに実行させてるので、Javaモード/Browserモードが実行されないことによるユーザの落胆を防ぐことができる。

以上が本発明に係る記録媒体の実施形態である。続いて本発明に係る再生装置の実施形態について説明する。図28は、本発明に係る再生装置の内部構成を示す図である。本図に示すように、再生装置は、BD-ROMドライブ1、トラックバッファ2、デマルチプレクサ3、ビデオデコーダ4、ピクチャプレーン5、オーディオデコーダ6、イメージメモリ7、イメージプレーン8、イメージデコーダ9、加算器10、静的シナリオメモリ11、再生制御エンジン12、プレーヤレジスタ13、BACKUP

メモリ 14、動的シナリオメモリ 15、DVD ライクモジュール 16、Java モジュール 17、BROWSER モジュール 18、U0 コントローラ 19、モジュールマネージャ 20、ディスパッチャ 21、レンダリングエンジン 22、通信部 23 から構成される。

5 BD-ROM ドライブ 1 は、BD-ROM のローディング／イジェクトを行い、BD-ROM に対するアクセスを実行する。

トラックバッファ 2 は、FIFO メモリであり、BD-ROM から読み出された ACCESS UNIT が先入れ先出し式に格納される。

デマルチプレクサ 3 は、トラックバッファ 2 から ACCESS UNIT を取り
10 出して多重分離を行い、GOP を構成するビデオフレームと、オーディオ
フレームとを得てビデオフレームをビデオデコーダ 4 に出力し、オーデ
ィオフレームをオーディオデコーダ 6 に出力する。副映像ストリームは
イメージメモリ 7 に格納し、Navigation Button 情報は動的シナリオメ
モリ 15 に格納する。デマルチプレクサ 3 による多重分離は、TS パケ
15 ットを PES パケットに変換するという変換処理を含む。

ビデオデコーダ 4 は、デマルチプレクサ 3 から出力されたビデオフレームを復号して非圧縮形式のピクチャをビデオプレーン 15 に書き込む。

ピクチャプレーン 5 は、非圧縮形式のピクチャを格納しておくための
メモリである。

20 オーディオデコーダ 6 は、デマルチプレクサ 3 から出力されたオーデ
ィオフレームを復号して、非圧縮形式のオーディオデータを出力する。

イメージメモリ 7 は、BD-ROM から読み出された副映像ストリーム、
Navigation Button 情報内の PNG データ、画像ファイルを格納しておく
バッファである。

25 イメージプレーン 8 は、一画面分の領域をもったメモリであり、展開
された副映像ストリーム、PNG データ、画像ファイルが配置される。

イメージデコーダ 9 は、イメージメモリ 7 に格納された副映像ストリーム、
PNG データ、画像ファイルを展開してイメージプレーン 8 に書き込む。
副映像ストリームのデコードにより、各種メニュー、副映像が画
30 面上に現れることになる。

加算器 10 は、ピクチャプレーン 5 に格納された非圧縮形式のピクチャデータに、イメージプレーン 8 に展開されたイメージを合成して出力する。図 13 (b) に示した画面(動画を表示した画面の横で CG(図中のフクロウの絵)が動きまわっているような画面)は、この加算器 10 が、
5 イメージプレーン 8 内のイメージと、ピクチャプレーン 5 内のピクチャとを合成することで出力される。

静的シナリオメモリ 11 は、カレントの PL やカレントのストリーム管理情報を格納しておくためのメモリである。カレント PL とは、BD-ROM に記録されている複数 PL のうち、現在処理対象になっているものをいう。カレントストリーム管理情報とは、BD-ROM に記録されている複数ストリーム管理情報のうち、現在処理対象になっているものをいう。
10

再生制御エンジン 12 は、AV 再生機能(1)、プレイリストの再生機能(2)、再生装置における状態取得／設定機能(3)といった諸機能を実行する。再生装置の AV 再生機能とは、DVD プレーヤ、CD プレーヤから踏襲した機能群であり、再生開始(Play)、再生停止(Stop)、一時停止(Pause
15 On)、一時停止の解除(Pause Off)、Still 機能の解除(still off)、速度指定付きの早送り(Forward Play(speed))、速度指定付きの巻戻し(Backward Play(speed))、音声切り換え(Audio Change)、副映像切り換え(Subtitle Change)、アングル切り換え(Angle Change)といった処理
20 をユーザからの操作に応じて実行することである。PL 再生機能とは、この AV 再生機能のうち、再生開始や再生停止を PL 情報に従って行うことをいう。この PL 再生機能の実行により、再生制御エンジン 12 はレイヤモデルの第 3 層(静的なシナリオに基づく再生制御)の役割を果たす。

再生制御エンジン 12 は、AV 再生機能をユーザからの操作に従って実行する。一方、機能(2)～(3)を、DVD ライクモジュール 16～BROWSER モジュール 18 からの関数呼出しに応じて実行する。つまり再生制御エンジン 12 は、ユーザ操作による指示、レイヤモデルにおける上位層からの指示に応じて、自身の機能を実行する。
25

プレーヤレジスタ 13 は、32 個の System Parameter Register と、
30 32 個の General Purpose Register とからなる。これらのレジスタの

格納値は変数 SPRM, GPRM としてプログラミングに利用される。System Parameter Register、及び、General Purpose Register は、DVD ライクモジュール 16～BROWSER モジュール 18 から分離した再生制御エンジン 12 で管理されるため、たとえ再生モードの切り換わりが生じたとしても、切換後の再生モードを実行するモジュールは、再生制御エンジン 12 における SPRM(0)～(31)、GPRM(0)～(31)を参照しさえすれば、再生装置の再生状態を知得することができる。

BACKUP メモリ 14 は、DVD ライクモジュール 16～BROWSER モジュール 18 の何れかが Suspend を実行した際に、再生装置レジスタの格納値の退避しておくためのスタックメモリである。この BACKUP メモリ 14 の格納値は、DVD ライクモジュール 16～BROWSER モジュール 18 の何れかが動的シナリオの Resume を実行した際、再生装置が具備するレジスタ格納値に復帰される。DVD ライクモジュール 16～BROWSER モジュール 18 のうち、同じものが 2 回以上 Suspended を行った場合、レジスタの格納値は、先入先出し式に格納される。このスタックの段数以上に格納値が格納された場合、それより前に退避された格納値が上書きされる。BACKUP メモリ 14 に格納される SPRM には、言語コード (Language Code)、現在デコード対象になっているオーディオストリームの番号 (Audio Stream Number)、現在デコード対象になっている副映像ストリームの番号 (Subtitle Stream Number)、ユーザにより設定されたアングル番号 (Angle Number)、現在再生中の Title の番号 (Title Number)、現在再生中の Chapter 番号、現在再生中の PL 番号 (PlayList Number)、現在再生中の CELL 番号 (PlayItem Number)、選択状態にあるボタンの番号 (Selected Button)、現在の再生時点を示す時刻情報といったものがある。

動的シナリオメモリ 15 は、カレント動的シナリオを格納しておき、DVD ライクモジュール 16～BROWSER モジュール 18 による処理に供されるメモリである。カレント動的シナリオとは、BD-ROM に記録されている複数シナリオのうち、現在実行対象になっているものをいう。

DVD ライクモジュール 16 は、MOVIE モードの実行主体となる DVD 仮

想プレーヤであり、動的シナリオメモリ 15 に読み出されたカレントの MOVIE オブジェクトを実行する。

Java モジュール 17 は、Java プラットフォームであり、Java 仮想マシン、コンフィグレーション、プロファイルからなる。Java モジュール 17 は、プログラム記録メモリ 1 に読み出された ZZZ.CLASS からカレントの Java オブジェクトを生成し、実行する。Java 仮想マシンは、Java 言語で記述された Java オブジェクトを、再生装置における CPU のネイティブコードに変換して、CPU に実行させる。

BROWSER モジュール 18 は、Browser モードの実行主体となるブラウザであり、動的シナリオメモリ 15 に読み出されたカレントの WebPage オブジェクトを実行する。BROWSER モジュール 18 が使用することができるプロトコルには、HTTP、IP、ARP、RARP、TCP、telnet、SMTP、ftp がある。DVD ライクモジュール 16～BROWSER モジュール 18 は何れも、再生装置に予め実装されているレジデントプログラムである。

UO コントローラ 19 は、リモコンや再生装置のフロントパネルに対してなされたユーザ操作を検出して、ユーザ操作を示す情報(以降 UO(User Operation)という)をモジュールマネージャ 20 に出力する。

モジュールマネージャ 20 は、BD-ROM から読み出された Index Table を保持して、モード管理及び分岐制御を行う。モジュールマネージャ 20 によるモード管理とは、動的シナリオをどの DVD ライクモジュール 16～BROWSER モジュール 18 に実行させるかという、モジュールの割り当てである。モジュール割り当ての原則は、動的シナリオを DVD ライクモジュール 16 に実行させるというものである。イントラモードでの分岐(同一モード内の分岐)があったとしても、この原則は維持される。例外は、インター モードでの分岐(モード間の分岐)が発生した場合である。MOVIE オブジェクトから Java オブジェクト／WebPage オブジェクトへの分岐が発生した場合、Java モジュール 17、BROWSER モジュール 18 がカレントオブジェクトを実行することになる。

モジュールマネージャ 20 による分岐制御は、分岐先となる動的シナリオを特定し、これをメモリに読み出して、DVD ライクモジュール 16

～BROWSER モジュール 18 のどれかに実行させることをいう。分岐先動的シナリオの特定は、間接参照形式で分岐先動的シナリオが指定されている場合に特に必要となる。この特定は、分岐コマンドの分岐先ラベルを参照し、これに対応する Index から、ファイル名を取り出すことでな
5 される。この特定にあたってモード切り換えの要否を併せて判定する。
モード切り換えの要否判定は、分岐先ラベルに対応する Index において、何というファイル拡張子が格納されているかを参照することでなされる。
分岐先動的シナリオが MOVIE モードなら拡張子は MOVIE、分岐先動的シ
ナリオが Java モードなら拡張子は CLASS、分岐先動的シナリオが
10 Browser モードなら拡張子は HTML, XML に設定されているので、Index に
格納された拡張子が何であるかで、モード切り換えが必要かどうかがわ
かる。モード切り換えが必要なら、分岐先動的シナリオをメモリに読み
出させて、切り換え後のモードを実行するモジュールに、モード遷移要
求を出力する。切り換え後のモードを実行するモジュールは、この遷移
15 要求の出力により、メモリ中の分岐先動的シナリオを実行することとな
る。

ディスパッチャ 21 は、U0 から、現在の再生装置におけるモードに
適切な U0 のみを選んで、そのモードを実行するモジュールに受け渡す。
例えば MOVIE モードの実行中に、上下左右、アクティベートといった
20 U0 を受け付けた場合、MOVIE モードのモジュールにこれらの U0 を出力
するというのがディスパッチャ 21 の処理である。これらの U0 は、
MOVIE モードにおけるメニュー挙動にのみ必要であり、Java モード／
Browser モードでは、不必要だからである。

レンダリングエンジン 22 は、Java3D, OPEN-GL といった基盤ソフト
25 ウェアを備え、Java モジュール 17 からの指示に従ってコンピュー
タ・グラフィックスの描画を行い、描画されたコンピュータ・グラフィ
ックスをイメージプレーン 8 に出力する。

通信部 23 は、BD-ROM からの指示に従って TCP/IP に基づく通信手順
を実行して、Web サイトへのアクセスを行う。
30 以上が再生装置の構成要素である。続いてモジュールマネージャ 20

についてより詳しく説明する。

モジュールマネージャ20は、汎用のCPUに、図29の処理手順を行うプログラムを読み取ることで実施可能である。図29は、モジュールマネージャ20による処理手順を示すフローチャートである。以降本フローチャートを参照しながら、モジュールマネージャ20による分岐制御について説明する。本フローチャートは、Index TableのFirstPlayのIndexからファイル名を取り出し(ステップS1)、カレントモードをMOVIEモードに設定して(ステップS2)、取り出されたファイル名の動的シナリオをカレントの動的シナリオに設定し(ステップS3)、カレントの動的シナリオ*i*を、メモリに読み出してから(ステップS4)、メモリ中のカレントの動的シナリオを実行するというものである(ステップS5～ステップS10)。

このステップS4～ステップS10は、カレントの動的シナリオが新たに設定される度に、実行される。

ステップS5～ステップS10は、シナリオを構成する各コマンドについて、ステップS6～ステップS10の処理を繰り返すというループ処理を形成している。本フローチャートにおける”*x*”とは、シナリオを構成するコマンドのうち、処理対象たるものを見つける変数である。このループ処理は、変数*x*を初期化し(ステップS5)、カレントの動的シナリオ*i*内のコマンド*x*を、カレントモードのモジュールに実行させてから(ステップS6)、ステップS7～ステップS8に規定された判定処理を行い、その後、変数*x*をインクリメントして(ステップS10)、ステップS6に戻るというものである。変数*x*をインクリメントしてステップS6に戻るという処理を繰り返すので、シナリオを構成する全てのコマンドに対し、ステップS6～ステップS10の処理は繰り返される。ループ処理の実行中、U0が発生した場合は(ステップS7)、U0のディスパッチ処理(図30のステップS31～ステップS33)を実行してからステップS8に戻る。U0のディスパッチ処理とは、コマンド実行中に発生したU0が上下左右、アクティベートであり(ステップS31)、カレントモードがMOVIEモードである場合に(ステップS32)、カレン

トモードのモジュールに U0 を出力するというものである。コマンド実行中に発生した U0 が上下左右、アクティベート以外の U0 なら、そのままカレントモードのモジュールに U0 を出力する(ステップ S 33)。コマンド実行中に発生した U0 が上下左右、アクティベート以外の U0 であり、カレントモードが MOVIE モードでないなら、発生した U0 は、モジュールに出力されない。以上がディスパッチ処理である。

この処理を行いつつ、ステップ S 6～ステップ S 10 の処理は繰り返される。このステップ S 6～ステップ S 10 のループ処理において処理対象たるカレント動的シナリオの切り換えは、ステップ S 8 にて Yes と判定された場合になされる。ステップ S 8 は、コマンド x が分岐コマンドであるかどうかの判定であり、このステップ S 8 が Yes ならば、ステップ S 11～ステップ S 20 においてカレントの動的シナリオを、新たな動的シナリオに設定してステップ S 4 に戻る。これにより、新たな動的シナリオがメモリに読み込まれて実行されることになる。

以降、ステップ S 11～ステップ S 23 の処理について説明する。この処理は、分岐制御であり、ステップ S 11、ステップ S 14、ステップ S 19、ステップ S 22 の判定結果に応じて、異なる処理を実行するものである。ステップ S 11 は、分岐コマンドの分岐先が Title ラベルを用いて記述されているか否かの判定であり、もし Yes ならステップ S 22 の判定を経て分岐先ラベルである Titlej を取得し(ステップ S 1 2)、Index Table における Titlej の Indexi からファイル名 j を取り出す(ステップ S 1 3)という処理を行う。もし No なら、分岐先を示すファイル名 j を取り出す(ステップ S 2 1)という処理を行う。

ステップ S 14 は、分岐コマンドが Call コマンドであるか、Jmp コマンドであるかの判定であり、Call コマンドなら、カレントの動的シナリオ i を Suspend して、変数 x を退避する(ステップ S 1 5)。Jmp コマンドなら、カレントの動的シナリオ i を discard する(ステップ S 1 6)。

以上の処理を経れば、ファイル名 j から特定される動的シナリオをカレントの動的シナリオ i に設定し(ステップ S 1 7)、取り出された拡張

子から特定される再生モードを再生モード k に設定する(ステップ S 1
8)。これらの設定後、ステップ S 19 を実行する。ステップ S 19 は、
再生モード k がカレントモードかどうかの判定であり、違うなら再生モ
ード k をカレントモードに設定してから(ステップ S 20)、ステップ S
5 4 に移行する。その後、新たに設定されたカレント動的シナリオに対し
て、ステップ S 4～ステップ S 10 の処理が繰り返されることになる。
尚ステップ S 22 は、再生装置がコアシステムであるか、フルシステム
であるかの判定であり、もしコアシステムであるなら、Title#0 の
Index からファイル名を取り出し、これを分岐先とする(ステップ S 2
10 3)。

このステップ S 4～ステップ S 10 のループ処理の終了要件は、ステ
ップ S 9において Yes と判定されることである。コマンド x が動的シナ
リオ i の最後のコマンドであるなら(ステップ S 9 が Yes)、動的シナリ
オ i の末尾に Resume コマンドがあるかどうかの判定がなされて(ステッ
15 プ S 34)、もしこれが No なら本フローチャートの処理を終了する。も
し動的シナリオ i の末尾に Resume コマンドが存在するなら、Suspend
中の動的シナリオを動的シナリオ i に設定し(ステップ S 35)、動的シ
ナリオ i のモードをカレントモードに設定して(ステップ S 36)、カ
レントモードのモジュールに、Suspend 中の動的シナリオ i の Resume を
20 行わせ(ステップ S 37)、変数 x を Suspend 前の値に戻してから(ステ
ップ S 38)、ステップ S 6～ステップ S 10 からなるループ処理のうち、
ステップ S 10 の直前に移行する。以上がモジュールマネージャ 2
0 の処理手順である。

また再生制御エンジン 12 は、汎用の CPU に、図 31 の処理手順を行
25 うプログラムを読み取ることで実施可能である。図 31 は、再生制御
エンジン 12 による PLPlay 関数の実行手順を示すフローチャートであ
る。本フローチャートにおいて処理対象たる PL を PLx、処理対象たる
CELL を CELLY、処理対象たる ACCESS UNIT を ACCESS UNITv とする。本
フローチャートは、PLPlay 関数の引数で指定された PL を PLx に設定し
30 (ステップ S 41)、PLx をメモリに読み込んでから(ステップ S 42)、

処理対象たる CELL を特定し(ステップ S 4 3～ステップ S 4 7)、以降、この CELL を構成する ACCESS UNIT を読み出す(ステップ S 4 8～ステップ S 5 1)という手順からなる。

ステップ S 4 3 は、CELL の引数指定が有るかどうかの判定であり、
5 もし引数指定があれば、CELLy を PLx の先頭 CELL に設定し(ステップ S
4 4)、CELLz を PLx の最後の CELL に設定する(ステップ S 4 5)。
CELLz とは、読み出範囲の最後を規定する CELL である。

引数指定がなければ、CELLy を引数指定された CELL に設定し(ステップ S 4 6)、CELLz を同じ引数指定された CELL に設定する(ステップ S
10 4 7)。CELLy、CELLz を、引数指定された CELL に設定しておくのは、
引数で CELL が指定されている場合は、その CELL さえ読み出せばよいからである。

ステップ S 4 8～ステップ S 5 3 は、CELLy を構成する ACCESS UNIT の読み出しと、復号手順とを示す。この手順は、CELLy の In 点ビデオ
15 フレームを含む ACCESS UNITv を TMAP から特定し(ステップ S 4 8)、
ACCESS UNITv の読み出しを BD-ROM ドライブ 1 に命じて(ステップ S 4
9)、ステップ S 5 0～ステップ S 5 1 の判定を経てから、ACCESS
UNITv に含まれるビデオフレームのデコードを行うようビデオデコーダ
20 4 に指示して(ステップ S 5 2)、ACCESS UNITv を次の ACCESS UNIT に
設定する(ステップ S 5 3)というものである。以降、CELLy に属する全
ての ACCESS UNIT について、このステップ S 4 9～ステップ S 5 3 の処
理が繰り返される。

ステップ S 5 0 は、ACCESS UNITv が In 点ビデオフレームを含んでいるかどうかの判定である。もし IN 点ビデオフレームを含んでいる場合
25 は(ステップ S 5 0 で Yes)、In 点のビデオフレームから ACCESS UNITv の最後のビデオフレームまでのデコードを行うようビデオデコーダ 4 に
指示してから(ステップ S 5 4)、ステップ S 5 2 に移行する。

ステップ S 5 1 は、Out 点 v が CELLy の Out 点ビデオフレームを含んで
30 いるかどうかの判定である。もし Out 点ビデオフレームを含んでいる場合は(ステップ S 5 1 で Yes)、ACCESS UNITv の先頭ビデオフレームか

ら Out 点ビデオフレームまでのデコードを行うようビデオデコーダ 4 に指示してから(ステップ S 5 5)、ステップ S 5 6 の判定を行う。ステップ S 5 6 は、本フローチャートの終了判定であり、CELLy が CELLz になったかを判定している。もしステップ S 5 6 が Yes なら本フローチャートを終了し、そうでないなら、CELLy を次の CELL に設定して(ステップ S 5 7)、ステップ S 4 8 に戻る。以降、ステップ S 5 6 が Yes と判定されるまで、ステップ S 4 8～ステップ S 5 7 の処理は繰り返される。以上が再生制御エンジン 1 2 の処理手順である。

以上のように本実施形態によれば、エンハンスドモードにあたっては、Java 仮想マシンやブラウザの特性を活かしたゲーム等を再生装置に実行させることができるので、動画データ本体の付加価値を高めることができる。またムービーモードからエンハンスドモードへの分岐は、テーブルを介した間接参照にて分岐先を指定しているので、テーブルの記述内容を工夫すれば、Java 仮想マシン、ブラウザをもつ再生装置への装填時と、もたない再生装置への装填時とで、分岐先を変えるという動作が実現可能になる。かかる分岐先の変化により、Java 仮想マシン、ブラウザをもたないコアシステム再生装置への装填時に、エンハンスドモードプログラムへの分岐の途を閉ざすことができ、あらゆるタイプの再生装置での動作保証を実現することができる。

20 (第 2 実施形態)

第 1 実施形態は、MOVIE オブジェクトにおいて設定されたレジスタ設定値を引き継いで、動作を行うような Java オブジェクトについて説明した。これに対し第 2 実施形態は、PL の再生を行い、この再生に同期して様々な再生制御を行う Java オブジェクトについて説明する。

25 図 3 2 は、第 2 実施形態に係る BD-ROM のファイル構成を示す図である。本図において新規なのは、YYY.Mark(PLMark), XXX.Mark(ClipMark) が追加されている点である。

PLMark は、PL の再生中において、再生装置が拡張制御を行うべき区間を示す情報である。YYY.Mark のファイル名 YYY は、PLMark が対応している PL と同じ名称が使用される。つまり本図における PLMark のファ

イル名は YYY であるから、PL(YYY, PL)に対応していることを意味する。

ClipMark は、AV ストリームの再生中において、再生装置が拡張制御を行うべき区間を示す情報である。XXX.Mark のファイル名 XXX は、

ClipMark が対応している AV ストリームと同じ名称が使用される。つまり本図における ClipMark のファイル名は XXX であるから、AV ストリーム(XXX.M2TS)に対応していることを意味する。

ClipMark と PLMark との違いは以下の通りである。ClipMark は、拡張制御を行うべき区間を AV ストリームに対して指定しているのに対し、PLMark は PL に対して指定している。

例えば、図 1 1 に示したように 1 つの AV ストリームに対し、2 つの PL 情報が定義されている場合に、ClipMark にて拡張制御を行うべき区間を指定すれば、この指定は、この AV ストリームを指定している 2 つの PL 情報の何れに対しても及ぶ。一方、PLMark にて拡張制御を行うべき区間を指定すれば、PLMark に対応する PL にのみ、拡張制御指定の効力は及ぶ。

PLMark に対応する PL が PL#1 なら PL#1 のみに及び、PL#2 には及ばない。つまり、拡張制御を行うべき区間を ClipMark にて指定すれば、AV ストリームを含む PL の全てに対し ClipMark の効力が及び、拡張制御を行うべき区間を PLMark にて指定すれば、その PLMark に対応する PL に対してのみ ClipMark の効力が及ぶのである。これが

ClipMark と PLMark との違いである。

本実施形態における拡張制御とは、再生装置にイベントを発生させるというものである。イベントを発生させるため、PLMark 及び ClipMark は、図 3 3 に示す共通のデータ構造をもつ。図 3 3 は、PLMark 及び ClipMark の共通構成を示す図である。本図に示すように PLMark は、イ

ベント数 (Number) 、個々のイベント (Event#1～Event#m) とから構成され、再生中に発生すべきイベントを定義する。個々のイベント (Event#) は、イベントの種類 (Type) と、イベントの ID (ID) と、イベントが発生する時刻 (Time) と、このイベントが有効となる時間帯の時間長 (Duration) を示す。

かかるデータ構造にて定義されるイベントには、TimeEvent、

UserEvent というものがある。TimeEvent とは、プレイリストの再生時間軸において、現在の再生位置が予め定められた時点 T に到達した場合に発生するイベントである。UserEvent とは、現在の再生位置が予め定められた時間帯に到達し、この時間帯でユーザにより操作がなされた場合に発生するイベントである。

プレイリスト #1 の再生中に出現する TimeEvent を定義する場合、PLMark をどのように記述するかについて図 34 (a) を参照しながら説明する。図中の下段の時間軸は、PL#1 が再生される時間軸を示す。この時間軸において時刻 t1 から発生する TimeEventEx1 を定義するケースを想定する。この TimeEvent を定義する場合の PLMark の記述例が、図中の PLMarkmt1 である。本 PLMark は、Type の項目が "TimeEvent" と記述され、ID の項目に "Ex1"、Time の項目が "t1"、Duration の項目が "0" と記述されている。図中の矢印は、時刻 t1 の到来で生じる TimeEvent を示す。この TimeEvent の発生により、図中のイベントandler が駆動されることになる。

一方、再生時間軸上の時刻 t1 から、間隔 T で TimeEvent を発生させる場合の PLMark の記述例を図 34 (b) に示す。図 34 (b) が (a) と違うのは、PLMark の Duration 項目が "T" と記述されている点である。項目 Duration の記述により、時点 t1 の経過後、間隔 T 置きに TimeEvent を発生させることができる。

プレイリスト #1 の再生中において UserEvent を定義する場合、PLMark をどのように記述するかについて図 35 を参照しながら説明する。図 35 の中段の時間軸は、PL#1 が再生される再生時間軸を示す。この時間軸上において現在の再生時点がどこであるかは、SPRM(10) に示されている。この時間軸において時刻 t1 から期間 T1 の間、有効となる UserEventEv1 を定義するケースを想定する。この UserEvent を定義した PLMark が、図中の PLMarkmt2 である。本 PLMark は、Type の項目が "UserEvent" と記述され、ID の項目に "Ev1"、Time の項目が "t1"、Duration の項目が "T1" と記述されている。本図におけるリモコン rml は、ユーザ操作を受け付ける機器であり、図中の矢印 uv1 は、

リモコンの ENTER キーの押下で生じる U0 を示す。U0 が、時刻 t1 から期間 T1 の時間帯に発生すれば、U0 に基づき UserEventEv1 が発生する。この UserEvent により、図中のイベントハンドラが駆動される。以上が PLMark により定義されるイベントである。尚 ClipMark により定義されるイベントについての説明は省略する。何故なら、PLMark が PL 再生中に発生すべきイベントを定義しているのに対し、ClipMark は AV ストリーム再生中に発生すべきイベントを定義しているにすぎず、PLMark におけるイベント定義とあまり変わらないからである。

続いて第 2 実施形態における Java オブジェクトについて説明する。
10 第 2 実施形態における ZZZ.CLASS のメンバー関数は PL 再生中に再生装置が発生するイベントにより駆動されるイベントハンドラである。以降、Java オブジェクトのメンバー関数たるイベントハンドラについて、具体例を交えながら説明する。図 3 6 は、TimeEvent により駆動されるハンドラの一例、図 3 7 は UserEvent により駆動されるハンドラの一例である。

図 3 5～図 3 6 における記述例では、以下の描画関数を用いて、イメージデータのデコードを行わせているものとする。

描画関数：PNG データをイメージプレーンに描画する

20 Draw (File, X, Y)

File : PNG データを格納したファイル名

X : X 座標位置

Y : Y 座標位置

25

イメージプレーンクリア関数：イメージプレーンの指定領域をクリアする

Clear (X, Y, W, H)

X : X 座標位置

30 Y : Y 座標位置

W : X 方向幅

H : Y 方向幅

図 3 6 におけるハンドラ hd1 は、 TimeEvent t1 が発生した際、

5 function{～}を実行するというハンドラである。 TimeEvent t1 とは、
PL#1 の再生開始直後(これを時間 t1 という)に発生するイベントである。
ハンドラ hd1 の function は、 GPRM(0)に 1 を設定し(GPRM(0)=1;)、 座標
(10, 200)に 1black.png を描画し(Draw(1black.png, 10, 200))、 座標
(330, 200)に 2white.png を描画する(Draw(2white.png, 330, 200))とい
う
10 ものである。 1black.png とは、 選択状態にあるボタンのイメージデー
タである。 2white.png とは、 通常状態にあるボタンのイメージデータ
である。 図中の画像イメージ ig1 は、 本図のイベントハンドラにより描
画される画像イメージである。 図中のボタン bn1, 2 は、 1black.png、
2white.png を復号することにより得られる。 GPRM(0)は、 この 2 つのボ
15 タンの何れが選択状態にあるかを示す。 この GPRM(0)が「1」に設定さ
れていることは、 ボタン bn1, 2 のうち、 ボタン bn1 が選択状態に設定さ
れていることを示す。

ハンドラ hd2 は、 TimeEvent t2 が発生した際、 function{～}を実行
するというハンドラである。 TimeEvent t2 とは、 PL#1 の再生終了直前
20 (これを時間 t2 という)に発生するイベントである。 ハンドラ hd2 の
function は、 PlayPL(-, -, 0)を実行するというものである。 この(-, -
, 0)は、「自身のプレイリストの自身 Cell の先頭から」という意味であ
る。

図 3 7 のハンドラ hd3 は、 PL#1 の再生中に UserEvent が発生した際、
25 駆動されるハンドラである。 ハンドラ hd2 の<event_handler id=
UserEvent k1>は、 UserEvent k1 の発生時に、 function 以下の内容を実
行することを示す。

function 以下の内容は、 2 つの if 文からなる。 1 つ目の if 文の条件
式は、 GPRM(0)が 1 であり、 尚且つ、 SPRM(8)が右であるという条件
30 (GPRM(0)==1 && SPRM(8)==Right) であり、 この条件が真である場合、

GPRM(0)に 2 を設定し(GPRM(0)=2;)、座標(10, 200)に 1white.png を描画し(Draw(1white.png, 10, 200))、座標(330, 200)に 2black.png を描画するという(Draw(1black.png, 330, 200))ものである。

図中のイメージ ig2 は、この条件成立時に描画される画像である。if 5 文の条件が真であったなら、イメージ ig2 のように、右側のボタン bn2 が選択状態に設定される。図 3 6 の画像イメージ ig1 を、その横に示す。この対比からわかるように、イベントハンドラは UserEvent の発生に応じて選択状態たるボタンを切り換えているのである。

if 文の else 節は、SPRM(8)が OK であるかという条件の成立を判定するものであり、この条件が成立すれば、2 つ目の if 文が実行されることになる。

2 つ目の if 文の条件は、(GPRM(0)==1) というものであり、この条件が真なら本 if 文は、プレイリスト #2 の Cell#1 の先頭から、再生を行う(PlayP1(PL#2, Cell#1, 0))。この条件が偽ならこの if 文は、プレイリスト #3 の Cell#1 の先頭から、再生を行う(PlayP1(PL#3, Cell#1, 0))。

図 3 8 は、図 3 6、図 3 7 のプレイリストにて再生されるピクチャと、イベントハンドラにより描画されるイメージとを合成して、イベントハンドラによる条件分岐を描いた図である。TimeEvent t1 により駆動されるイベントハンドラは、ボタンを合成するというものなので、プレイリストにおける最初のピクチャ pc1、途中のピクチャ pc2、最後のピクチャ pc3 に対し、イメージ ig1(ボタン bn1, 2 を含む)が合成されて表示される。TimeEvent t2 により駆動されるイベントハンドラは、プレイリストの Cell の最後のピクチャデータが再生された後、先頭に戻るというものだから、最後のピクチャデータの表示後、矢印 py1 に示すように最初のピクチャデータが表示され、これにボタンが合成される。ユーザからの操作がない限り、このような合成画像の表示が繰り返される。ユーザが矢印キーを押下することで UserEvent が発生したなら、矢印 ky1 に示すようにイメージ ig1 の代わりにイメージ ig2 が、この AV ストリームを構成する各ピクチャに合成される。

30 イメージ ig1 が表示された状態でユーザが Enter キーを押下すれば、

矢印 ey1 に示すようにイベントハンドラ hd3 内の PlayP1(PL#2, Cell#1, 0) が実行され、プレイリスト #2 の Cell#1 を再生させる。

5 イメージ ig2 が表示された状態でユーザが Enter キーを押下すれば、 イベントハンドラ hd3 内の PlayP1(PL#3, Cell#1, 0) が実行され、矢印 ey2 に示すようにプレイリスト #3 の Cell#1 を再生させる。

ユーザの操作があるまで 1 つの Cell を繰り返し再生し、ユーザの操作があれば、ボタンに応じた分岐先に分岐するという再生手順が、イベントハンドラを用いて簡単に記述できる。

10 かかるイベントハンドラにより、DVD で実現されているメニュー挙動と同様の制御を、プログラマが使い馴れた Java 言語を用いて簡単に記述することができる。更にこれを発展させれば、ボタンの代わりに CG が動作するような凝ったメニュー表示の実現が可能になる。これにより、映画制作の表現の幅が広がる。

15 以上が本実施形態における Java オブジェクトの改良である。続いて、本実施形態における再生装置の改良点について説明する。上述したような TimeEvent、UserEvent を発生させるには、再生制御エンジン 12 に図 39 に示すような処理手順を行わせればよい。

20 図 39 は、本実施形態に係る再生制御エンジン 12 の処理手順を示す図である。本フローチャートにおいて新規なのは、CELL を構成する ACCESS UNIT を読み出し、復号するという一連の処理中(ステップ S 4 9 ~ ステップ S 53)に、ステップ S 60、ステップ S 62 という 2 つの判定ステップが挿入されている点である。ステップ S 60 は、PL 再生において、現在の再生時点を発生時刻とした TimeEvent が、PLMark、 ClipMark に定義されているかどうかを判定するステップである。ステップ S 60 が Yes なら、TimeEvent を発生して(ステップ S 61)、ステップ S 51 に移行する。これにより、TimeEvent を駆動要件とするイベントハンドラが駆動される。

25 ステップ S 62 は、U0 が発生したかどうかを判定しているステップである。ステップ S 62 が Yes なら、PLMark、ClipMark を参照し、そ

の U0 を有効としている時間帯であるかを判定する(ステップ S 6 3)。もしその時間帯であれば、UserEvent の発生して(ステップ S 6 4)、ステップ S 5 1 に戻る。

以上のように本実施形態によれば、再生制御エンジン 1 2 が PL を再生させている間、再生時間の進行に応じてイベントを発生するので、5 Java モードにおいては、TimeEvent、UserEvent を駆動要件とするイベントハンドラを駆動することができる。これにより、動画の中身に密接に同期したゲームを、Java 言語を用いて記述することができる。

(第 3 実施形態)

10 第 2 実施形態は、PLMark、ClipMark で定義されたイベントにて駆動されるイベントハンドラについて説明したが、第 3 実施形態は、PLMark、ClipMark で定義されたメンバー関数の挙動を、AV 再生機能と同期させる場合の改良に関する。再生装置の AV 再生機能とは、DVD プレーヤ、CD プレーヤから踏襲した機能群であり、再生開始(Play)、再生停止
15 (Stop)、一時停止(Pause On)、一時停止の解除(Pause Off)、Still 機能の解除(still off)、速度指定付きの早送り(Forward Play(speed))、速度指定付きの巻戻し(Backward Play(speed))、音声切り換え(Audio Change)、副映像切り換え(Subtitle Change)、アングル切り換え(Ungle Change)といったものがある。

20 ユーザがこれらの機能の実行を命じた場合、再生装置は実行すべき機能に応じたイベントを発生する。巻戻し機能の実行をユーザが命じたなら Backward Event、早送り機能の実行をユーザが命じたなら Fastward イベント、一時停止機能の実行をユーザが命じたなら Pause Event を再生装置は発生する。これらのイベントでイベントハンドラを駆動するのは、以下の意義がある。再生という動作は、PL の再生時間軸上でなされる。一方、Java 仮想マシンの動作は時間軸という概念をもたない。時間軸を前提にしていないので、PL の再生制御と、Java 仮想マシンの動作とを同期させるには、PL の再生がどこまで進行したかを何等かの術を用いて Java 仮想マシンに伝えねばならない。時間軸における PL の
25 再生制御を Java 仮想マシンに伝えるイベントこそ、Backward Event、
30 Backward Event、

TimeEvent、Pause Event である。

(Backward Event)

Backward Event とは、巻戻し操作がユーザによりなされた場合、巻戻し操作がなされたという事実にてイベントハンドラを駆動するための 5 イベントである。この Backward Event を導入したことの技術的意義は以下の通りである。図 4 0 は、Backward Event の技術的意義を説明するための模式図である。この図 4 0 において、Java オブジェクトのメンバー関数がキャラクタ A の CG を画面上で動作させるプログラムであるものとする。このメンバー関数を、PL 再生開始から時間 t1 後に駆動 10 したい場合、時間 t1 経過という TimeEvent を PLMark に定義し、上述したプログラムをこの TimeEvent により駆動されるイベントハンドラとして記述すればよい。

かかる TimeEvent による駆動で問題となるのは、「巻戻しては進む」「巻戻しては進む」という変則的な再生を続けた場合である。この場合、15 再生開始から時刻 t1 が経過したという TimeEvent が何度も発生することになり、そうすると本来 1 つでよい筈の CG が、複数画面に現れるという事態が生じる。図 4 0 における bg1 は、再生が時刻 t1 まで到来したためにハンドラにより描画される CG を示す。また bg2 は、巻戻しにより再生が時刻 t1 まで到来したためにハンドラにより描画される CG を 20 示す。本来 1 つでよい筈の CG が 2 つ描画されるため、画面デザインがおかしくなってしまう。これを避けるには、時間 t1 を経過し、その後の巻戻しで再度時間 t1 に到来した際、CG を消すような動作を Java オブジェクトに記述しておくのが望ましい。そこで巻戻し操作がなされた場合、再生装置に Backward Event(図中の bw1)を発生させ、これにより 25 駆動されるイベントハンドラを ZZZ.CLASS に定義しておくのである。巻戻し操作がなされ、再度 TimeEvent t1 が発生した場合、画面上の CG を消す処理をかかるイベントハンドラに行わせることにより、「巻戻しては進む」「巻戻しては進む」という変則的な再生を繰り返した場合でも、画面に現れる CG は一体のみとなる。

30 図 4 1、図 4 2 は、Backward Event により駆動されるイベントハン

ドラを用いた場合の画面表示例である。図 4 1 (a) は、Java モードにおいて、PL 再生が開始した時点の表示画面の一例であり、図 4 1 (b) は、PL の再生時点が時刻 t_1 に到達した場合の表示画面の一例である。時刻 t_1 への到達により TimeEvent が発生し、この TimeEvent によりイベントハンドラが駆動される。図中のフクロウは、Java オブジェクトのイベントハンドラが駆動したことにより描画されるコンピュータ・グラフィックスである。

図 4 1 (c) は、PL の再生時点が時刻 t_2 に到達した場合の表示画面の一例である。時刻 t_2 に到達した時点でユーザが巻戻しを行ったものとする。図 4 2 (a) は、巻戻しにより PL の再生時点が時刻 t_1 に到達した場合の表示画面の一例である。時刻 t_1 への到達により、図中のフクロウを消去する。これは上述したようなフクロウの二重描画を防止するためである。以降巻戻しにより再生時点が t_0 にまで到達し(図 4 2 (b))、その後、ユーザが再度通常再生を行ったものとする。図 4 2 (c) は、PL の再生時点が時刻 t_1 に再到達した場合の表示画面の一例である。巻戻し時にコンピュータ・グラフィックスを消去したため、時刻 t_1 への到達にて CG を再描画したとしても、画面に現れる CG は唯一つになる。

尚本実施形態では、Backward Event にて CG の消去を行ったが、 t_2 にて発生する TimeEvent を PLMark に定義して、

TimeEvent t_1 の発生 → TimeEvent t_2 の発生 → TimeEvent t_1 の発生

という順序で TimeEvent が発生すれば、フクロウのコンピュータ・グラフィックスを消去してもよい。

(TimeEvent)

第 2 実施形態で説明した TimeEvent は、早送り再生がなされた際、早送りの速度に応じた時間間隔で、発生するという性質をもつ。図 4 3 は、

早送り時での Time Event 発生を模式的に示す図である。本図において TimeEvent は、時間間隔 drl を開けて発生している。この時間間隔は、ユーザによる早送り速度の指定に応じたものであり、PL の再生速度が早ければ早い程間隔は短く、再生速度が遅ければ遅い程、間隔は長くなる。
5 よって TimeEvent は、PL の再生速度が早ければ早い程、短い間隔で発生する。逆に PL の再生速度が遅ければ遅い程、長い時間間隔で発生する。このイベント発生の時間間隔が長いか短いかで Java オブジェクトは、PL の再生が早く進んでいるか、遅く進んでいるかを知ることができる。そのため PL の再生が早く進んでいる場合と、遅く進んでいる場合とで、CG の動きを変えるようプログラムを行うこともできる。
10 これにより、CG の動きと、PL 再生とに緻密な時間同期をとらせることができる。

図 4 4 は、早送りに応じた TimeEvent により駆動されるイベントハンドラを用いた場合の画面表示例である。図 4 4 (a) は、Java モードにおいて、PL 再生が開始した時点の表示画面の一例であり、図 4 4
15 (b) は、PL の再生時点が時刻 t1 に到達した場合の表示画面の一例である。時刻 t1 への到達により TimeEvent が発生し、この TimeEvent によりイベントハンドラが駆動される。図中のフクロウは、このイベントハンドラが駆動したことにより描画されるコンピュータ・グラフィックスである。図 4 4 (c) は、早送りを行った表示画面の一例である。早送りがなされたことより、TimeEvent の発生間隔が変わる。Java オブジェクトは、この間隔変化に応じて図中のコンピュータ・グラフィックスの動きを変える。
20

図 4 4 (c) におけるフクロウの飛翔は、コンピュータ・グラフィックスの描画方法を変化させる一例である。TimeEvent の発生間隔に応じてコンピュータ・グラフィックスの描画方法を変化させることにより、PL が早送りにて再生された場合でも、Java オブジェクトの動きを緻密に同期させることができる。尚図 4 4 における画面の横線は、早送りなされている様子を、VTR の再生に例えて誇張して表現したものにすぎない。実際の BD-ROM での早送り時には、このような横線は画面には現
30

れない。ここで 10 倍速のように、高速である場合は、コンピュータ・グラフィックスを消去してしまうのが望ましい。

(ポーズ)

Pause Event は、一時停止操作がなされた際、再生装置において発生するイベントである。図 45 は、Pause Event の発生を模式的に描いた図である。かかる Pause Event の発生時において、再生装置の動作を止めるようイベントハンドラを記述すれば、PL の再生は止まっているのに、CG の動きだけは進行しているというような齟齬を無くすことができる。

以上のイベントにて、動作するようなイベントハンドラをメンバー関数として定義することで、Java オブジェクトの動作が PL 再生と緻密に同期するような高品位な映画作品を、視聴者に供給することができる。

以上のように本実施形態によれば、再生時間軸において現在の再生時点がどこであるか、再生がどれだけのスピードに応じて進行しているかによって、Java オブジェクトの処理を変化させてるので、Java オブジェクトにより描画される CG の動作のリアリティを高めることができる。

(第 4 実施形態)

本実施形態は、DVD と同様のメニュー制御を BD-ROM 上で実現する場合の改良に関する。図 46 は、BD-ROM により実現されるメニュー階層を示す図である。本図におけるメニュー階層は、TopMenu を最上位に配し、この TopMenu から下位の TitleMenu、SubTitleMenu、AudioMenu を選択できる構造になっている。図中の矢印 sw1, 2, 3 は、ボタン選択によるメニュー切り替えを模式的に示す。TopMenu とは、音声選択、字幕選択、タイトル選択の何れを行うかを受け付けるボタン(図中のボタン sn1, sn2, sn3)を配置したメニューである。

TitleMenu とは、映画作品(Title)の劇場版を選択するか、ディレクターズカット版を選択するか、ゲーム版を選択するか等、映画作品の選択を受け付けるボタンを配置したメニューである。AudioMenu とは、音声再生を日本語で行うか、英語で行うかを受け付けるボタンを配置した

メニュー、SubTitleMenu とは、字幕表示を日本語で行うか、英語で行うかを受け付けるボタンを配置したメニューである。

こうした階層をもったメニューを動作させるための MOVIE オブジェクトを図 47 に示す。

5 FirstPlay オブジェクト(FirstPlay OBJ)は、再生装置への BD-ROM のローディング時におけるスタートアップ手順を記述した動的シナリオである。FirstPlay オブジェクトを表す四角枠は、このスタートアップ手順を実行するためのコマンドを示す。FirstPlay オブジェクトの最後のコマンドは分岐コマンドであり、TopMenu オブジェクトを分岐先にして
10 いる。

TopMenu オブジェクト(TopMenu OBJ)は、TopMenu の挙動を制御する動的シナリオである。TopMenu オブジェクトを表す四角の枠は、この制御手順を現した個々のコマンドを模式化したものである。このコマンドには、ユーザからの操作に応じて TopMenu 中のボタンの状態を変えるものや、ボタンに対する確定操作に応じて分岐を行う分岐コマンドがある。
15 この分岐コマンドは、TopMenu から TitleMenu、TopMenu から

SubTitleMenu、TopMenu から AudioMenu というメニュー切り換えを実現するものである。

AudioMenu オブジェクト(AudioMenu OBJ)は、AudioMenu の挙動を制御する動的なシナリオである。AudioMenu オブジェクトを構成する四角の枠は、この制御手順を現した個々のコマンドを模式化したものである。このコマンドには、ユーザからの操作に応じて AudioMenu 中のボタンの状態を変えるものや、ボタンに対する確定操作に応じて音声設定用の SPRM を更新するコマンドがある。

25 SubTitleMenu オブジェクト(SubTitleMenu OBJ)は、SubTitleMenu の挙動を制御する動的なシナリオである。SubTitleMenu オブジェクトを構成する四角の枠は、この制御手順を現した個々のコマンドを模式化したものである。このコマンドには、ユーザからの操作に応じて SubTitleMenu 中のボタンの状態を変えるものや、ボタンに対する確定
30 操作に応じて音声設定用の SPRM を更新するコマンドがある。

TitleMenu オブジェクト (TitleMenu OBJ) は、TitleMenu の挙動を制御する動的シナリオである。TitleMenu オブジェクトを表す四角の枠は、この制御手順を現した個々のコマンドを模式化したものである。このコマンドには、ユーザからの操作に応じて TitleMenu 中のボタンの状態を 5 変えるものや、ボタンに対する確定操作に応じて分岐を行う分岐コマンドがある。この分岐コマンドは、個々の Title への分岐を実現するものである。

これらのメニュー用 MOVIE オブジェクトにより、DVD で実現されているようなメニューの挙動を実現することができる。以上がメニュー制御 10 に関連する MOVIE オブジェクトである。

続いて本実施形態における Index Table の改良について説明する。本実施形態における Index Table には、FirstPlayINDEX、Audio MenuINDEX、Subtitle MenuINDEX、Title MenuINDEX が追加されている。第 1 実施形態に示したように、これらの Index は、3 つのモードの何れ 15 の動的シナリオからも参照される。

First PlayINDEX は、BD-ROM のスタートアップ時に参照される Index である。FirstPlay オブジェクトのファイル名が本 Index に記述される。

Top MenuINDEX、Audio MenuINDEX、Subtitle MenuINDEX、Title MenuINDEX は、それぞれ Audio Menu、Subtitle Menu、Title Menu をダイレクトにコールするようなユーザ操作がなされた際、参照される Index である。ユーザによるダイレクトなコールは、ユーザがリモコン 20 における Audio 選択キー、Subtitle 選択キー、Title 選択キーを押下することにより場合になれる。

以上が本実施形態における MOVIE オブジェクトの改良である。続いて、 25 本実施形態における再生装置の改良点について説明する。上述したような MOVIE オブジェクトを動作させるには、モジュールマネージャ 20 に 図 48 のフローチャートに示すような処理手順を行わせればよい。

メニュー制御を行うべく、本実施形態を示すモジュールマネージャ 20 は、図 48 に示すような処理手順で分岐制御を行う。本フローチャートが新規なのは、ステップ S 8 とステップ S 11 との間にステップ S 2 30

4 が挿入されており、このステップ S 24 が Yes である場合に、ステップ S 25～ステップ S 28 の処理を行って、ステップ S 4 に戻る点である。ステップ S 25～ステップ S 28 は、メニューの制御を行うシナリオを、カレント動的シナリオに設定するものである。つまり分岐コマンドの分岐先が xxxMenu であるなら(ステップ S 24 で Yes)、カレントの動的シナリオ i を Suspend し、変数 x を退避して(ステップ S 25)、分岐先 Menu に対応する Index からファイル名を取り出し(ステップ S 26)、取り出されたファイル名の動的シナリオをカレントの動的シナリオ i に設定し(ステップ S 27)、カレントモードを MOVIE モードに戻す(ステップ S 28)。その後、カレント動的シナリオを実行してゆく。

以上のように本実施形態によれば、メニュー制御のための動的シナリオへの分岐は、Index Table の Index を介した間接参照にて実現されるので、Java モード、Browser モードの実行時においてメニューキーが押下された場合でも、これらメニュー制御用の動的シナリオへと分岐することができる。Java 仮想マシン、Browser モードからの音声切り換え、字幕切り換えの実現が可能となり、Java 仮想マシン、Browser モードでの再生にあたっても、通常の DVD 同様、音声切り換え、字幕切り換えが可能となる。

(第 5 実施形態)

20 第 1 実施形態において、Java モードのアプリケーションプログラムは、再生制御エンジンが提供するプログラミング関数、システム変数を用いて BD-ROM に対する再生制御を記述していた。つまり再生制御の記述に、再生制御エンジンが提供するプログラミング関数、システム変数をダイレクトに用いていた。これに対し第 5 実施形態における Java モードのアプリケーションは、Java モジュール 17 が提供するメンバー関数を介して BD-ROM に対する制御を記述している。

どのようなメンバー関数を介して記述するかについて、図 49 を参照しながら説明する。図 49 は、Java モードに属するクラスのメンバー関数を示す図である。第 4 層にあたる Java モードは、BD-ROMStatus クラス、BD-ROMPlayBack クラス、BD-ROMEvent クラスを含む BD-ROM 専用

パッケージを有している。

以降、パッケージにおける個々のクラスについて説明する。

BD-ROMEvent クラスは、setEventListener 関数を始め複数のメンバー関数を含む。setEventListener 関数は、ZZZ.CLASS のインスタンス (Java オブジェクト) が生成された際、Java オブジェクトのメンバー関数の使用宣言を行うものである。つまり Java オブジェクトのメンバー関数は、この setEventListener 関数の使用宣言があって初めて使用が可能となる。TimeEvent や UserEvent が発生した場合、

setEventListener 関数にて使用宣言がなされたイベントハンドラが駆動される。

BD-ROMPlayBack クラスは、PlayPL at CELL()、PlayPL at Mark()、PlayPL at Specified Time() をメンバー関数としてパッケージングしたクラスである。

BD-ROMStatus クラスは、状態取得や状態設定用のメンバー関数、つまり、第 1 実施形態に示した (i) Get value of Player Status Register 関数、(ii) Set value of Player Status Register 関数、(iii) Set value of Player Status Register 関数、(iv) Get value of General Purpose Register 関数、(v) Set value of General Purpose Register をメンバー関数としてパッケージングしたクラスである。

図中の矢印 ycl, 2, 3, 4, 5 は、BD-ROMStatus クラス、BD-ROMPlayBack クラス、BD-ROMEvent クラスのメンバー関数を介した PL 再生機能、レジスタ設定値の取得／設定を模式的に示す。この矢印に示すように、第 5 実施形態における Java オブジェクトは、Java モジュール 17 におけるパッケージを介して、再生制御エンジン 12 が提供するプログラミング関数、レジスタ設定値であるプレーヤ変数を利用しているのである。

図 50 は、Java モード内のメンバー関数を介した制御を示す図である。

矢印 yp1 は、ZZZ.CLASS のインスタンスたる Java オブジェクトの生成を象徴化している。この生成は、Java オブジェクトへの分岐が最初に命じれた際、Java モジュール 17 によりなされる。矢印 yp2 は、

Java オブジェクトによる setEventListener 関数の呼出を象徴化している。

矢印 yh1 は、イベントオブジェクト内の setEventListener 関数によるイベントハンドラの使用宣言を象徴化して示す。矢印 yh2 は、イベントオブジェクト内のメンバー関数によるイベントハンドラの駆動を象徴化して示す。つまり JAVA オブジェクトのイベントハンドラは、イベントオブジェクトのメンバー関数 (setEventListener 関数) からイベントハンドラ使用宣言を受けて使用可能になり、ClipMark、PLMark により定義された KeyEvent、TimeEvent にて駆動される。

矢印 yh3 は、駆動されたイベントハンドラによる状態取得を象徴化している。つまり第 5 実施形態における Java オブジェクトのメンバー関数たるイベントハンドラは、Java モードに属するクラスのメンバー関数 (get System Parameter Register) を用いて間接的に再生装置の状態取得や状態設定を行う。

矢印 yh4 は、駆動されたイベントハンドラによるプレイリスト再生を象徴化している。つまり Java オブジェクトのメンバー関数たるイベントハンドラは、BD-ROMPlayback クラスのメンバー関数を用いてプレイリストの再生を装置に命じるのである。

以上のように本実施形態によれば、BD-ROM に対する再生制御を、Java モジュール 17 におけるパッケージを介して記述するので、一般的の Java 言語によるプログラミングスタイルと同じスタイルで、BD-ROM に対する再生制御を記述することができる。これにより、映画作品の制作に参入したソフトハウスの制作効率を高めることができる。

尚、本実施形態のパッケージは、BD-ROM 規格のライセンス管理を行う団体と、アプリケーションプログラムを開発するソフトハウスとが正式な契約を締結することを条件として、この団体により交付されるようになることが望ましい。この契約は、BD-ROM に記録された映画作品の品位を汚すようなアプリケーションプログラムを開発しないという禁止条項を含んでいる。このようにすれば、映画作品の品位を汚すという恐れを避けつつ、様々なアプリケーションをソフトハウスに開発させるこ

とができる。

(第6実施形態)

第1実施形態～第5実施形態において MOVIE オブジェクトは、Navigation Button 情報における分岐コマンドに基づき MOVIE モードから Java モードへの移行を行ったが、本実施形態では、メニューを介して MOVIE モードから Java モードへの移行を行う。図51は、第6実施形態に係るメニュー階層を示す図である。図46のメニュー階層画像と異なるのは、TopMenu から Extra Menu への移行が可能である点である。Extra Menu とは、Java モード／Browser モードの何れを選択するかをユーザから受け付けるメニューであり、Java モードへの移行を受け付けるボタン、Browser モードへの移行を受け付けるボタンを含む。図52は、第6実施形態に係る MOVIE オブジェクト、Index Table を示す図である。本図と図47との違いは、Extra Menu の挙動を制御する ExtraMenu オブジェクト(ExtraMenu OBJ)が存在し、TopMenu の MOVIE オブジェクトから、この Extra Menu オブジェクトへの分岐が可能である点(1)、Index Table 内に Extra Menu の Index があり、リモコンのメニューキーの押下により Extra Menu オブジェクトへの分岐が可能である点(2)である。

メニューキーの押下により、Java モードへと移行するので、現在の再生時点がどこであろうと、Java モードへの移行が行われることになる。Browser モードへの移行も同様である。

以上が、本実施形態における MOVIE オブジェクト、Index Table の改良である。続いて本実施形態における Java オブジェクトの改良について説明する。本実施形態における Java オブジェクトは、MOVIE モードから Java モードへと移行した際、SPRM(10)を参照して、MOVIE モードにおいてどの再生時点から Java モードへと分岐したかについて知得する。そして SPrM(10)と、予め具備しているスケジュールテーブルと(図53)を照合して、Java モードにおける処理を行う。このスケジュールテーブルは、動画データに登場しているキャラクタと、このキャラクタが登場している時間帯とを対にして示すテーブルである。

Java オブジェクトはこのテーブルを参照して、現在の再生時点にあるキャラクタを知得し、このキャラクタを主人公にしたゲームを実行する。

以上のように本実施形態によれば、メニューを介した分岐により、
5 MOVIE モードから Java モードへと移行することができる。また、
分岐直前の再生時点を示す SPRM(10)を参照することで、ユーザがどこ
まで視聴したかに応じて、処理を切り換えることができる。かかる切り
換えで、映画作品の再生と緻密に関連するような Java 言語アプリケー
ションを制作することができる。

10 (第 7 実施形態)

第 1 実施形態は、MOVIE モードで再生させていた PL 内の CELL を Java
モードからも再生させるという PL のシェアリングが可能であった。し
かしこのシェアリングは、PL 情報に、フィルタリング情報を設ける場
合に、障害となる。フィルタリング情報とは、AV ストリームに多重化
15 されている複数ストリームのうち、どれを有効とし、どれを無効とする
かを定める情報である。フィルタリング情報を設ける理由は、以下の通
りである。AV ストリームには、映像・音声以外にも、Navigation
Button 情報や副映像ストリームというようなストリームが含まれてい
る。Navigation Button 情報は、MOVIE モードにあたっては必要である
20 が、Java モードや Browser モードではそうとは限らない。何故なら
Java モードでは、Java 言語でコンピュータ・グラフィックスを描画す
ることができる、Navigation Button 情報の力を借りるまでもないからで
ある。Java モードにおいては、Navigation Button 情報を無効と扱うた
め、かかるフィルタリング情報が必要となる。ところがこのフィルタリ
25 ング情報を PL 情報に設けて、上述したような PL のシャアリングがなさ
れれば、ストリームの有効、無効を定めるフィルタリング情報もシェア
リングされてしまう。

フィルタリング情報を PL 情報に設けた場合、PL 情報のシェアリング
は望ましくないので、本実施形態では図 54 のように PL を設定する。
30 図 54 は、第 7 実施形態に係る PL 情報の構成を示す図である。本図の

PL 情報には、フィルタリング情報と App.Flag とが新規に追加されている。

App Flag は、プレイリストをどのモードの動的シナリオに利用させるかを指定する情報である。アプリケーションプログラムの指定は、

5 MOVIE モード、Java モード、Browser モードという 3 者の中から 1 つを選ぶことでなされる。この App.Flag が「00:MOVIE モード」を示しているなら、MOVIE オブジェクトならどこでもこの PL 情報を利用することができます。しかし Java オブジェクト、MOVIE オブジェクトは、利用することはできない。

10 一方 App.Flag が「01:Java モード」を示しているなら、Java オブジェクトであればどちらでもこの PL 情報を引用することができる。

「10:Browser モード」を示しているなら、「WebPage オブジェクト」が引用することができる。更に「11」を示しているのであればどちらでもこの PL 情報を引用することができる。そして 11 を示しているなら、どのモードの動的シナリオであっても、このプレイリスト情報を引用することができる。

App.Flag により、1 つの PL 情報を 1 つのモードの動的シナリオに利用させ、他のモードの動的シナリオに利用させないという排他制御を実現することができる。かかる排他制御により、PL 情報に設定されたフィルタリング情報は、本来用いられるべきモードの動的シナリオにより使用されることになり、意図しないモードでのストリーム再生を防ぐことができる。

PL 情報を設けることで、図 55 のようなシェアリングが可能となる。

図 55 は、デジタルストリーム本体を MOVIE モードと、Java モードとでシェアリングしつつ、PL 情報を MOVIE モードと、Java モードとで排他的に用いているという階層的なシェアリングを示す図である。

App.Flag に基づく処理を行うため、本実施形態に係る動的シナリオは、PlayPL 関数の呼出しに先立ち、再生したい PL の App.Flag を再生制御エンジン 12 から受け取る。そして動的シナリオは、自身が属するモードと、受け取った App.Flag に示されるモードとが一致するかを判定

し、一致していれば PlayPL 関数の呼出を行う。一方不一致ならば関数呼出を行わない。動的シナリオ及び再生制御エンジン 1 2 が以上の処理を行うことにより、App.Flag に示されるモードの動的シナリオだけが、PL を再生することになる。

5 尚、PlayPL 関数の関数呼出時に、動的シナリオは自身が属するモードを再生制御エンジン 1 2 に通知し、再生制御エンジン 1 2 はそのモードが関数呼出先の PL の App.Flag と一致しているかどうかを判定してもよい。そして一致していれば、その PlayPL 関数を実行し、不一致なら実行しないのである。こうすることで、App.Flag に示されるモードの動的シナリオだけが、PL を再生することになる。

本実施形態に示したような階層的なシェアリングにより、フィルタリング情報を活用しつつ、MOVIE モードにおけるワンシーンを Java モードで利用するというシーン引用が可能になる。

無論、フィルタリング情報や App.Flag をストリーム管理情報に設けてもよい。しかしフィルタリング情報をストリーム管理情報に設けると、モード間の AV ストリームのシェアリングが一切不可能になってしまい、賢明な策とはいがたい。他のモードでの引用を実現をしようとするのなら、PL 情報に App.Flag、フィルタリング情報を設けて、AV ストリーム本体をシェアリングできる途を残しておくことが望ましい。

20 (第 8 実施形態)

本実施形態は、BD-ROM の製造工程に関する実施形態である。図 5 6 は、第 8 実施形態に係る BD-ROM の製造工程を示すフローチャートである。

BD-ROM の制作工程は、動画収録、音声収録等の素材作成を行う素材制作工程 S 1 0 1、オーサリング装置を用いて、アプリケーションフォーマットを生成するオーサリング工程 S 1 0 2、BD-ROM の原盤を作成し、プレス・貼り合わせを行って、BD-ROM を完成させるプレス工程 1 0 3 を含む。

これらの工程のうち、BD-ROM を対象としたオーサリング工程は、シナリオ編集工程 S 2 0 1、素材エンコード工程 S 2 0 2、多重化工程 S

203、フォーマッティング工程 S 204、エミュレーション工程 S 205という5つの工程とからなる。

シナリオ編集工程 S 201とは、企画段階において作成された筋書きを再生装置が理解できる形式に変換する工程である。シナリオ編集の結果は、BD-ROM 用シナリオとして生成される。また、このシナリオ編集において、多重化を実現するため多重化パラメータの等も生成される。

素材エンコード工程 S 202とは、ビデオ素材、オーディオ素材、副映像素材のそれぞれをエンコードして、ビデオストリーム、オーディオストリーム、副映像ストリームを得る作業である。

10 多重化工程 S 203では、素材エンコードにより得られた、ビデオストリーム、オーディオストリーム、副映像ストリームをインターリープ多重して、これらを1本のデジタルストリームに変換する。

15 フォーマッティング工程 S 204では、BD-ROM 向けシナリオを元に、各種情報を作成して、シナリオ及びデジタルストリームを BD-ROM のフォーマットに適合させる。

エミュレーション工程 S 205では、オーサリング作業の結果が正しいか否かの確認を行う。

上述したシナリオ編集工程において、Java オブジェクト及び WebPage オブジェクトは、Java 言語、マークアップ言語を用いた記述が可能であるから、通常のコンピュータ向けのソフトウェアを開発するのと同じ感覚で、開発することができる。よって本実施形態では、このシナリオ制作の効率を高めることができるという効果がある。

(第9実施形態)

第9実施形態は、本発明に係る記録媒体を BD-RE 等の書き換え可能な光ディスクとして実施する場合の実施形態である。第1実施形態に示した AV ストリーム-ストリーム管理情報-PL 情報といったデータ構造は、BD-RE と互換を有しており、BD-RE の記録装置は、放送信号のリアルタイムレコーディングを行うことにより、AV ストリーム-ストリーム管理情報-PL 情報といったデータ構造を BD-RE に書き込む。これらが書き込まれた BD-RE に対して、ユーザがプログラミングを行い、MOVIE オ

プロジェクト、Java オブジェクト、WebPage オブジェクト、Index Table を記述して BD-RE に書き込めば、第 1 実施形態に示したアプリケーションフォーマットと同様のフォーマットを、BD-RE 上で実現することができる。

5 以上のように本実施形態によれば、BD-RE の記録装置を所持するユーザに、第 1 実施形態に示したアプリケーションフォーマットを作成する機会を与えるので、多くのユーザに、第 1 実施形態に示した映画作品を制作する楽しさを知ってもらうことができる。

(第 10 実施形態)

10 第 10 実施形態は、Java オブジェクト、WebPage オブジェクトの動作が可能であったとしても、一定の条件の下に再生装置をコアシステムに固定化してしまう実施形態である。

つまり以下の事情(1)～(7)が再生装置側にある場合、ムービーモードからエンハンスドモードへの遷移の途を閉ざすのである。

15 映画作品の著作権保護のための暗号鍵がある再生装置で暴露され、再生装置が鍵管理センターにより無効化された場合(1)、

ユーザがリッパソフトを利用して記録媒体に記録された映画作品を不正にコピーしたため、映画作品の複製物がネットワーク上に或はされる恐れがある場合(2)、

20 エンハンスドモード実行には追加料金の支払いがいるにも拘らず、追加料金が支払われない場合(3)、

再生装置のシステムに障害が発生したため、ネットワークから遮断したい場合(4)、

25 Java 仮想マシンやブラウザにバージョンコンフリクションがある場合(5)、

個人情報の漏洩やウィルスソフトの感染の可能性があり、再生装置をネットワークから遮断したい場合(6)、

ネットワークから記録媒体の記録内容を読み取ろうとする不正機器から、記録媒体の映画作品を保護するため、再生装置をネットワークから遮断したい場合(7)、

本実施形態における Java オブジェクト、WebPage オブジェクトは以上の事情(1)～(7)の有無をチェックするようなチェックルーチンをもっている。このチェックルーチンは、再生装置における不正ソフトウェアのインストールやウィルスの感染等の再生装置に対するチェックをネットワーク上のサーバと連携して実行する。このサーバは、再生装置による暗号鍵利用を管理する暗号鍵管理センタ、ユーザに対する課金を行う課金センタ、不正コピー摘発を推進する著作権管理センタ等により運営されるものであり、Java オブジェクト、WebPage オブジェクトがこれらのセンタのサーバと通信を行い、(1)～(7)の条件が再生装置側に満たされれば、強制的にコアシステムに移行するよう、Java オブジェクト、WebPage オブジェクトはプログラミングされている。

ここでリッパソフトを用いた不正コピーの可能性の有無は、Java オブジェクト、WebPage オブジェクトにチェックルーチンを組み込むことで実現される。このチェックルーチンは、HD のファイル、フォルダを定期的にセンターに報告し、センタがこれを検閲するという処理で実現される(尚、かかる検閲は映画作品購入時の事前承認が前提である。)。

以上のように本実施形態によれば、ネットワークを介した高付加価値化を実現しつつ、著作権者や再生装置の製造メーカーが何等かの事情でネットワーク接続を閉ざすことを希望した場合、再生装置をコアシステムに設定し再生装置をスタンドアローンで利用するよう Java オブジェクト、WebPage オブジェクトを記述することができる。特に、再生装置に HD やオペレーティングシステムがインストールされ、パソコンのような再生環境で BD-ROM が利用されている場合に特に有意義になる。尚上述した事情の発生後、適切な処置がとられれば、フルシステムに移行するようにしてもよい。

(第 1 1 実施形態)

第 1 1 実施形態は、再生装置に HD をもたせて Index Table と、新たな PL とを通信部 2 3 にダウンロードさせて HD に書き込ませる実施形態である。HD に新たな Index Table が書き込まれれば、PL から PL への分岐にあたってモジュールマネージャ 2 0 は、BD-ROM に記録された Index

Table の代わりに HD に記録された Index Table からファイル名を取り出し、そのファイル名の動画データを読み出させて DVD ライクモジュール 16～BROWSER モジュール 18 に実行させる。

以上のように本実施形態によれば、BD-ROM に記録された特定の PL に
5 道徳・倫理的な問題がある場合、Index Table と、差し替え映像とを再生装置にダウンロードさせれば、新たな Index Table を介した間接参照を再生時に再生装置に行わせることで、問題がある映像の代わりに、差し替え映像をユーザに視聴させることができる。一部差し替えにあたって BD-ROM に記録された動的シナリオを一切書き換えなくてもよいので、
10 かかる問題が生じたとしても、記録媒体を回収するというリスクを避けることができる。

また差し替え映像でなくても、BD-ROM に記録されている複数 Title のうち、特定のものを再生させたくない場合、BD-ROM に記録されている Title の順序に入れ替えたい場合も、Index Table をダウンロードするだけでよい。何故なら、これらの変更にあたって、BD-ROM に記録されている動的シナリオの書き換えは不要だからである。
15

(備考)

上記実施形態に基づいて説明してきたが、現状において最善の効果が期待できるシステム例として提示したに過ぎない。本発明はその要旨を
20 逸脱しない範囲で変更実施することができる。代表的な変更実施の形態として、以下(A)(B)(C)……のものがある。

(A)全ての実施形態では、本発明に係る光ディスクを BD-ROM として実施したが、本発明の光ディスクは、記録される動的シナリオ、Index Table に特徴があり、この特徴は、BD-ROM の物理的性質に依存するもの
25 ではない。動的シナリオ、Index Table を記録しうる記録媒体なら、どのような記録媒体であってもよい。例えば、DVD-ROM, DVD-RAM, DVD-RW, DVD-R, DVD+RW, DVD+R, CD-R, CD-RW 等の光ディスク、PD, MO 等の光磁気ディスクであってもよい。また、コンパクトフラッシュカード、スマートメディア、メモリスティック、マルチメディアカード、PCM-CIA カード等の半導体メモリカードであってもよい。フレキシブルディスク、
30

SuperDisk, Zip, Clik! 等の磁気記録ディスク(i)、
ORB, Jaz, SparQ, SyJet, EZFley, マイクロドライブ等のリムーバルハード
ディスクドライブ(ii)であってもよい。更に、機器内蔵型のハードディ
スクであってもよい。

5 動的シナリオ、Index Table、プレイリスト情報は、AVストリーム及
びストリーム管理情報と別々の記録媒体に記録されてもよい。そしてこ
れらをパラレルに読み出して、1つの映像編集として再生させてもよい。

(B) 全ての実施形態における再生装置は、BD-ROMに記録されたAVス
トリームをデコードした上でTVに出力していたが、再生装置をBD-ROM
10 ドライブのみとし、これ以外の構成要素をTVに具備させてもい、この
場合、再生装置と、TVとをIEEE1394で接続されたホームネットワーク
に組み入れることができる。また、実施形態における再生装置は、テレ
ビと接続して利用されるタイプであったが、ディスプレイと一体型とな
った再生装置であってもよい。更に、各実施形態の再生装置において、
15 処理の本質的部分をなす部分のみを、再生装置としてもよい。これらの
再生装置は、何れも本願明細書に記載された発明であるから、これらの
何れの態様であろうとも、第1実施形態～第11実施形態に示した再生
装置の内部構成を元に、再生装置を製造する行為は、本願の明細書に記
載された発明の実施行為になる。第1実施形態～第11実施形態に示
20 した再生装置の有償・無償による譲渡(有償の場合は販売、無償の場合は贈
与になる)、貸与、輸入する行為も、本発明の実施行行為である。店頭展
示、カタログ勧誘、パンフレット配布により、これらの譲渡や貸渡を、
一般ユーザに申し出る行為も本再生装置の実施行行為である。

(C) 図29～図31、図39、図48に示したプログラムによる情報
25 処理は、ハードウェア資源を用いて具体的に実現されていることから、
上記フローチャートに処理手順を示したプログラムは、単体で発明とし
て成立する。全ての実施形態は、再生装置に組み込まれた態様で、本發
明に係るプログラムの実施行行為についての実施形態を示したが、再生裝
置から分離して、第1実施形態～第11実施形態に示したプログラム單
30 体を実施してもよい。プログラム單体の実施行行為には、これらのプログ

ラムを生産する行為(1)や、有償・無償によりプログラムを譲渡する行為(2)、貸与する行為(3)、輸入する行為(4)、双方向の電子通信回線を介して公衆に提供する行為(5)、店頭展示、カタログ勧誘、パンフレット配布により、プログラムの譲渡や貸渡を、一般ユーザに申し出る行為(6)がある。

(D)図29～図31、図39、図48のフローチャートにおいて時系列に実行される各ステップの「時」の要素を、発明を特定するための必須の事項と考える。そうすると、これらのフローチャートによる処理手順は、再生方法の使用形態を開示していることがわかる。各ステップの処理を、時系列に行うことで、本発明の本来の目的を達成し、作用及び効果を奏するよう、これらのフローチャートの処理を行うのであれば、本発明に係る記録方法の実施行為に該当することはいうまでもない。

(E)Javaオブジェクトは、CGを描画するようなアプリケーションであったが、Java言語で記述されるアプリケーションであればどのようなものであってもよい。例えば電子商取引(EC(Electronic Commerce))のクライアントアプリケーションであってもよい。映画作品の動画を交えながら商品案内を行うようなJavaオブジェクトを実現することができるので、映画作品に関連するキャラクタビジネスを成功に導くことができる。またJavaオブジェクトのアプリケーションは、ネット対戦型のオンラインゲームであってもよい。更に、JavaオブジェクトにてCGにて表現されるキャラクタは、エージェントとしての処理を行うものであってもよい。エージェントとしてのキャラクタは、再生装置のヘルプ機能を実現するものであったり、ユーザに助言を行うものでもよい。

Javaオブジェクトが用いるようなライブラリをBD-ROMに記録してもよい。そのようなライブラリには、PNGファイル、アニメーションデータを格納したMNGファイル、ストリームに関連した情報を格納したXMLファイル、HTML/SMILファイルがある。特に、アニメーションデータを格納したMNGファイルがライブラリとして記録されている場合、上述したようなクロウのコンピュータ・グラフィックスを簡単に描画することができる。

WebResponse オブジェクトが WEB サイトから取得する情報は、WEB ページであってもよいし、画像データであってもよい。また、AV ストリーム、ストリーム管理情報、PL 情報であってもよい。また WebPage オブジェクトは、検索エンジンと連携して処理を行ってもよい。

5 更に、エンハンスドモードにおける記述言語は、C++や C# 言語であってもよい。

(F) 尚第 1 実施形態における図 18 の一例は、本 BD-ROM における再生制御の記述技法の一例にすぎない。他の記述技法としては、AV ストリーム中の Navigation Button 情報からダイレクトに Java オブジェクト 10 へと分岐するという記述技法がある。図 57 は、AV ストリーム中の Navigation Button 情報からダイレクトに Java オブジェクトへと分岐するという再生制御の一例である。かかる分岐を行うよう Navigation Button 情報を記述すれば、あるキャラクタが登場しているシーンから、そのキャラクタを主役としたゲームに分岐するというシーン展開を、自 15 由に記述することができるので、シーン展開の表現の幅が広がる。

(G) 第 5 実施形態では、Chapter を一覧表示するための Menu(Chapter Menu)と、この挙動を制御する MOVIE オブジェクトとを BD-ROM に記録しておき、Top Menu から分岐できるようにしてもよい。またリモコン 20 キーの Chapter キーの押下により呼出されるようにしてもよい。

(H) BD-ROM に記録するにあたって、AV ストリームを構成する各 TS パケットには、拡張ヘッダを付与しておくことが望ましい。拡張ヘッダは、TP_extra_header と呼ばれ、『Arrival_Time_Stamp』と、『copy_permission_indicator』とを含み 4 バイトのデータ長を有する。 25 TP_extra_header 付き TS パケット(以下 EX 付き TS パケットと略す)は、32 個毎にグループ化されて、3 つのセクタに書き込まれる。32 個の EX 付き TS パケットからなるグループは、6144 バイト($=32 \times 192$)であり、これは 3 個のセクタサイズ 6144 バイト($=2048 \times 3$)と一致する。3 個のセクタに収められた 32 個の EX 付き TS パケットを "Aligned Unit" とい 30 う。

IEEE1394 を介して接続されたホームネットワークでの利用時において、再生装置 200 は、以下のような送信処理にて Aligned Unit の送信を行う。つまり送り手側の機器は、Aligned Unit に含まれる 32 個の EX 付き TS パケットのそれぞれから TP_extra_header を取り外し、TS パケット本体を DTCP 規格に基づき暗号化して出力する。TS パケットの出力にあたっては、TS パケット間の隨所に、isochronous パケットを挿入する。この挿入箇所は、TP_extra_header の Arribval_Time_Stamp に示される時刻に基づいた位置である。TS パケットの出力に伴い、再生装置 200 は DTCP_Descriptor を出力する。DTCP_Descriptor は、
TP_extra_header におけるコピー許否設定を示す。ここで「コピー禁止」を示すよう DTCP_Descriptor を記述しておけば、IEEE1394 を介して接続されたホームネットワークでの利用時において TS パケットは、他の機器に記録されることはない。

(I) AV ストリームにコピー世代情報(CGI)を埋め込んで、AV ストリームの制限付きコピーを認めてよい。コピー世代情報には、自由にコピー可能な旨を示す『copy free』と、1 世代の複製物の記録が許可されている旨を示す『one generation copy』と、複製物の更なる複製は禁止されている旨を示す『no more copy』と、一切のコピーを認めない旨を示す『never copy』とがある。つまり、コンテンツに埋め込まれたコピー世代情報が『copy free』や『one generation copy』を示している場合のみ、HD から DVD へのバックアップを行い、『no more copy』『never copy』を示している場合、バックアップは行わないである。

(J) AV ストリームに権利管理情報は付加されている場合、権利管理情報に定められた利用条件において、AV ストリームのコピーを行ってよい。

権利管理情報において回数指定で、コピーの許可条件が規定されているなら、この条件内でコピーを行う。有効期間の指定(年月日の指定)がでコピーの条件が規定されているのならこの条件内でコピーを行う。

(K) コピー処理に、移動やマイグレイト、チェックアウトといったバリエーションが存在する場合、これらバリエーション毎に、定められた

利用条件に基づきバックアップを行っても良い。移動とは、コピー元のコンテンツの削除が伴うコピー処理であり、複数記録媒体間でコンテンツを移動する場合に用いられる。

マイグレイトとは、コピー先記録媒体において、利用条件情報を生成
5 させることを前提にしたコピー処理である。

チェックアウトとは、回数制限されたコピーの一種であり、コピー回数を1回デクリメントした上でコンテンツのコピーが実行される。一般にいう回数制限付きのコピーと、チェックアウトとの違いは、コピー回数のインクリメントが可能である点である。コピー回数のインクリメントは、コピーにより記録媒体に記録されたコンテンツを再生不能にするという処理("チェックイン"と呼ばれる。)を行った後でなされる。

(L)各実施形態におけるAVストリームは、DVD-Video規格、DVD-Video Recording規格のVOB(Video Object)であってもよい。VOBは、ビデオストリーム、オーディオストリームを多重化することにより得られたISO/IEC13818-1規格準拠のプログラムストリームである。また、各実施形態におけるAVストリームを、"AVClip"と呼んでもよい。この場合、ストリーム管理情報はClip情報と呼ぶことになる。またAVストリームにおけるビデオストリームは、MPEG4やWMV方式であってもよい。更にオーディオストリームは、Linear-PCM方式、Dolby-AC3方式、MP3方式、MPEG-AAC方式であってもよい。

(M)各実施形態におけるCell情報は、時間情報により再生区間の開始点、終了点を指定していたが、BD-ROMにおける論理アドレスを用いて再生区間の開始点、終了点を指定してもよい。また、各実施形態におけるCELLを"PlayItem"と呼んでもよい。

(N)ストリーム管理情報におけるTMAPは、EP_mapと呼んでもよい。この場合、ACCESS UNITの再生開始時刻を、ACCESS UNIT先頭に位置するピクチャデータのタイムスタンプ(Presentation Time Stamp)で表現するのが望ましい。またACCESS UNITにおけるアドレスは、PESパケットの連番(SPN(Serial Packet Number))で表現するのが望ましい。

(O)再生装置の構成において、動的シナリオメモリ15には、カレン

トの動的シナリオのみを格納し、静的シナリオメモリ 1 1 には、カレントのストリーム管理情報、カレントの PL 情報のみを格納するとしたが、キャッシュメモリのように、複数のシナリオ、ストリーム管理情報、PL 情報を予め格納しておいてもよい。こうすれば、これらのデータを BD-ROM から読み出すまでのタイムラグを短くすることができる。また BACKUP メモリ 1 4 は、スタック式にレジスタの格納値を退避しておくとしたが、メモリ規模との関連を考えると、退避すべき格納値は一段とするのが現実的である。

(P) PL を構成する 2 以上の CELL を連續再生させるには、これらの CELL がシームレス接続されよう、加工を施しておくことが望ましい。

シームレス接続のための加工は、動画データにおいて先行する側の再生区間の終端部と、後続する側の再生区間の先端部とを複製することにより、予め複製部分を作成しておき、これらを再エンコードすることで、実現される。尚、シームレス接続のために作成された複製部分を、 Bridge-Clip と呼んでもよい。

ここで終端部、先端部は、以下のように設定するのが望ましい。
つまり先行する VOB#x のうち先行再生区間の Out 点を含む ACCESS UNIT から、2 個先の ACCESS UNIT までを終端部とし、また後続する CELL 情報#x+1 のうち後続再生区間の In 点を含む ACCESS UNIT を先端部とするのが望ましい。終端部及び先端部をこのように定める根拠は、同出願人の先行技術米国特許 USP, 6148, 140 公報により記載されているので、詳細に関してはこの公報を参照されたい。

更に、シームレス接続のために作成された複製部分については、シームレス接続情報を設けておくことが望ましい。シームレス接続情報とは、最初のビデオフレームの再生開始時刻、最後のビデオフレームの再生終了時刻、オーディオギャップの開始時刻、オーディオギャップの時間長、オーディオギャップの位置情報を含む情報である。かかるシームレス接続情報が定義されていれば、最初のビデオフレームの再生開始時刻、最後のビデオフレームの再生終了時刻から、両区間のタイムスタンプの差 (STC-Offset) を計算して、再生装置に設定することができる。また、こ

これらオーディオギャップの情報を参照して、オーディオデコーダを制御すれば、1つの区間から別の区間への移行する際の音声の途切れを防止することができる。

(Q)各実施形態における Title(映画作品)は、テレビ映画、ゲームソフト等、映像をもって表現されている全ての著作物を意味する。何故なら、各実施形態における映像編集は、ブラウン管や液晶での表示など、視覚的又は視聴覚的效果を生じさせる方法で表現され(i)、BD-ROMという有体物に、何らかの方法で結びつくことにより同一性を保ちながら存続しており(ii)、かつ再現が可能な状態におかれているので(iii)、映画の著作物の存在形式としての要件を満たすからである。

しかし本発明の本質的要素は、この映像編集であることに限定されるものではないので、本発明の動画データは、監視カメラの映像や家庭用ビデオカメラの映像であってもよい。

(R)各実施形態における BD-ROM は、基礎出願に記載された HD-DVD, BD-ROM のことをいう。また各実施形態における再生制御エンジン 12 は、基礎出願の明細書に開示された BD-ROM-FF プロセッサ 3025、シナリオプロセッサ 304 (図 35) に該当する。またモジュールマネージャ 20 は、プレイバックコントローラ 3024 (図 35) に該当し、DVD ライクモジュール 16 ~ BROWSER モジュール 18 は、DVD 互換モジュール 3021、ブラウザモジュール 3022、Java モジュール 3023 が該当する。これらは基礎出願に記載されたものであるから、これらの技術事項についての優先権主張は、正当なものである。

(S)Java 言語により描画されるコンピュータ・グラフィックス(フクロウの絵等)は、NURBS(Non Uniform Rational B-Spline)形式のデータで作成されてもよい。NURBS とは、ベジェ曲線の束であって(このようなベジェ曲線の束は B-Spline と呼ばれる。)、それぞれのベジェ曲線の曲率が不均一なものをいう。

またポリゴン形式で作成されてもよい。ポリゴン形式とは、多面体近似により特定の立体形状を表現するよう規定されたデータ形式であり、アメリカ・オートキャド社が規定した Data eXchange Format(DXF)を初め、

HRC, WAVEFRONT, IV, VRML 形式等のものが広く知られている。

また形状表面の模様については、テクスチャマッピングにより付されてもよい。レンダリングエンジン 22 は、上述したような三次元形状データの投影像を、作成してイメージデコーダにデコードさせる。この際 5 レンダリングエンジン 22 は、テクスチャマッピングを行う。テクスチャマッピングとは、平面又は曲面に、静止画、ビットマップ等のテクスチャパターンを貼り付けて表示させる処理をいう。また光源位置と、各三次元形状データにおける各平面との距離や位置関係に基づいて制御点間の平面の色、明るさを算出して、この明るさにより、スクリーン上の 10 投影像の色・明るさを調整する。このように、光源位置と、各三次元形状データにおける各平面との距離や位置関係に基づいて、制御点間の平面の色、明るさを算出する処理はシェーディング処理といい、このようなシェーディング処理がなされることにより、三次元形状データの投影像には、陰影が付され、投影像に立体感が現れる。以上のような処理にて、専用のゲームマシン並みのコンピュータ・グラフィックスが、映画 15 の動画像の脇に表示されることになる。

フクロウの飛翔等のキャラクタの動きは、三次元形状データを一定の規則性に基づき変化させることでなされる。三次元形状データは、XXX.M2TS を構成するエクステントの合間(図 7 の『他のファイル』にある箇所)に記録しておくことが望ましい。こうすれば AV ストリームと、三次元形状データとをまとめて読み出せる。

(T) 各実施形態では、ユーザによる映像編集の選択操作を、リモコンから受け付けたが、再生装置のフロントパネルから受け付けてもよい。キーボード、タッチパネル、マウスやパッド、トラックボール等の入力機器にてユーザの指定を受け付けてもよい。この場合は、クリック操作、ドラッグ操作でこれらの操作を受け付けてもよい。

(U) 各実施形態における映像編集は、アナログ放送で放送されたアナログ映像信号をエンコードすることにより得られたものでもよい。デジタル放送で放送されたトランスポートストリームから構成されるストリームデータであってもよい。

またビデオテープに記録されているアナログ／デジタルの映像信号をエンコードしてコンテンツを得ても良い。更にビデオカメラから直接取り込んだアナログ／デジタルの映像信号をエンコードしてコンテンツを得ても良い。他にも、配信サーバにより配信されるデジタル著作物でもよい。

(V) Java モジュール 17 は、衛星放送受信のために機器に組み込まれた Java プラットフォームであってもよい。Java モジュール 17 がかかる Java プラットフォームであれば、本発明に係る生成装置は、MHP 用 STB としての処理を兼用することになる。

更に携帯電話の処理制御のために機器に組み込まれた Java プラットフォームであってもよい。かかる Java モジュール 17 がかかる Java プラットフォームであれば、本発明に係る生成装置は、携帯電話としての処理を兼用することになる。

また BROWSER モジュール 18 は、MicroSoft 社の Internet Explore 等、パソコン組み込み型のブラウザソフトであってもよい。

(W) 図 12 に示したレイアモデルにおいて、Java モードの上にブラウザモード及び MOVIE モードを配置してもよい。特に MOVIE モードでの動的シナリオの解釈や、動的シナリオに基づく制御手順の実行は、再生装置に対する負担が軽いので、MOVIE モードを Java モード上で実行させても何等問題は生じないからである。また再生装置や映画作品の開発にあたって、動作保証が 1 つのモードで済むからである。

更に 3 つのモードを設けず、Java モードだけで Java モードの処理を実行してもよい。第 2 実施形態に示したように、Java モードでも PL の再生と同期した再生制御が可能になるから、強いて MOVIE モードを設けなくてもよいという理由による。更に動的シナリオにおける制御は、MOVIE モードだけでも、ブラウザモードだけでもよい。

符号の説明

- 1 BD-ROM ドライブ
- 2 トランクバッファ
- 3 デマルチプレクサ

- 4 ビデオデコーダ
- 5 ピクチャプレーン
- 6 オーディオデコーダ
- 7 イメージメモリ
- 5 8 イメージプレーン
- 9 イメージデコーダ
- 10 10 加算器
- 11 11 静的シナリオメモリ
- 12 12 再生制御エンジン
- 10 13 プレーヤレジスタ
- 14 14 BACKUP メモリ
- 15 15 動的シナリオメモリ
- 16 16 DVD ライクモジュール
- 17 17 Java モジュール
- 15 18 BROWSER モジュール
- 19 19 UO コントローラ
- 20 20 モジュールマネージャ
- 21 21 ディスパッチャ
- 22 22 レンダリングエンジン
- 20 23 通信部
- 24 24 ストリームース
- 100 100 BD-ROM
- 200 200 再生装置
- 300 300 テレビ
- 25 304 304 シナリオプロセッサ
- 400 400 リモコン

産業上の利用可能性

本発明に係る記録媒体は、映画作品を構成する動画データの付加価値
30 を効果的に高めるので、より魅力ある映画作品を市場に供給することが

でき、映画市場や民生機器市場を活性化させることができる。故に本発明に係る記録媒体、再生装置は、映画産業や民生機器産業において高い利用可能性をもつ。

請求の範囲

1. 動画データ、複数のプログラム、テーブルが記録された記録媒体であって、

各プログラムは、

5 動画データの再生制御手順を示し

テーブルは、記録媒体における各プログラムの識別情報(1)、各プログラムがムービーモード及びエンハンスドモードの何れに属するかを示す情報(2)を含んでおり、

何れか1つのプログラムは、分岐コマンドを含み、

10 前記分岐コマンドは、テーブルを介した間接参照の形式で、分岐先を指定している

ことを特徴とする記録媒体。

2. 前記テーブルは、複数プログラムのそれぞれに対応した複数のインデックスを含み、

15 各インデックスは、

対応するプログラムがエンハンスドモードに属するか、ムービーモードに属するかを示しており、

前記分岐コマンドにおける間接参照の形式とは、

20 インデックスについてのラベルを用いて、分岐先プログラムを指定することである

ことを特徴とする請求項1記載の記録媒体。

3. 前記テーブルにおけるインデックスには、予備のインデックスがあり、

予備のインデックスは、

25 ムービーモードしか実行できない再生装置において、エンハンスドモードのプログラムへの分岐が命じられた場合に、エンハンスドモードのプログラムの代わりとなるムービーモードプログラムと対応していることを特徴とする請求項2記載の記録媒体。

4. 前記各ムービーモードのプログラム及びエンハンスドモードのプログラムのそれぞれは、2以上の実行モジュールのそれぞれにより実行

され、

2 以上の実行モジュールは、共通の制御階層上のレジデントプログラムであり、

前記再生制御手順は、

5 当該制御階層から供給される関数を用いて記述されていることを特徴とする請求項 1 記載の記録媒体。

5. 前記制御関数から供給される関数は、

予め定められた再生経路に基づく再生制御を再生装置に行わせる関数
(1)、

10 再生装置におけるレジスタに所定の値を設定する関数(2)、

再生装置におけるレジスタの設定値を取得する関数(3)のうち何れかである

ことを特徴とする請求項 4 記載の記録媒体。

6. 前記分岐とは、ムービーモードプログラムからエンハンスドモードプログラムへの分岐であり、

ムービーモードプログラムによる再生制御手順とは、再生装置のレジスタへの値の設定であり、

エンハンスドモードプログラムによる再生制御手順とは、レジスタに設定された値を、参照した処理である

20 ことを特徴とする請求項 5 記載の記録媒体。

7. 前記レジスタ設定値とは、

再生装置における音声設定(1)、再生装置における副映像設定(2)、再生装置におけるアングル設定(3)、現在再生されているタイトル(4)、現在再生されているチャプター(5)、現在の再生時点(6)の何れかを示す値である

ことを特徴とする請求項 6 記載の記録媒体。

8. 前記記録媒体にはプレイリスト情報が記録されており、

プレイリスト情報は、動画データにおける再生区間を示す情報を、再生順序に従って配列することにより再生経路を定義する情報であり、

30 制御階層から供給される関数による再生制御は、

プレイリスト情報に示される再生経路に基づくことを特徴とする請求項 5 記載の記録媒体。

9. 前記プログラムは、制御階層から供給される再生制御関数を呼び出すための関数呼出を含み、

5 この関数呼出には、2つの引数があり、

そのうち1つ目の引数は、プレイリスト情報を指定するものであり、2つ目の引数は、再生経路における開始点を指定するものであることを特徴とする請求項 8 記載の記録媒体。

10. 前記再生手順における開始点指定は、

再生区間、再生時刻、チャプターの何れかを用いて行われることを特徴とする請求項 9 記載の記録媒体。

11. 前記記録媒体においてプレイリスト情報は、フラグが付加されて記録されており、

15 フラグは、本プレイリスト情報を用いた動画データの再生を、ムービーモードで許可するか、エンハンスドモードで許可するかを示すことを特徴とする請求項 8 記載の記録媒体。

12. 前記分岐とは、ムービーモードプログラムからエンハンスドモードプログラムへの分岐であり、

ムービーモードプログラムによる動画データの再生手順とは、

20 プレイリスト情報により示される再生経路における開始点を指定して、再生を行わせることであり、

エンハンスドモードプログラムによる動画データの再生手順とは、同じプレイリスト情報により示される再生経路における開始点を指定して、再生を行わせることである

25 ことを特徴とする請求項 8 記載の記録媒体。

13. 前記エンハンスモードとは、仮想マシンに、プログラムを実行させるモードであり、

エンハンスドモードのプログラムは、仮想マシン向けプログラミング言語により記述される

30 ことを特徴とする請求項 1 記載の記録媒体。

14. 前記エンハンスドモードのプログラムにおいて、

前記再生制御手順は、動画データの再生中において、再生装置で発生するイベントをトリガにして実行される

ことを特徴とする請求項13記載の記録媒体。

5 15. 前記イベントとは、

現在の再生位置が、動画データの再生時間軸における所定の時点に到達したことを示すイベント(i)、

動画データの再生時間軸において、所定の時間間隔だけ、現在の再生位置が進行したことを示すイベント(ii)の何れかである

10 ことを特徴とする請求項14記載の記録媒体。

16. 前記記録媒体はマーク情報が記録されており、

マーク情報は、

前記再生時間軸における所定の時点、及び、所定の時間間隔を定義する情報である

15 ことを特徴とする請求項15記載の記録媒体。

17. 前記イベントとは、

動画データの再生時間軸上の所定の時間帯において、ユーザ操作がなされたことを示すイベントである

ことを特徴とする請求項13記載の記録媒体。

20 18. 前記ユーザ操作とは、

動画データの再生時において、早送り、巻戻し、ポーズが命じられたことを示す

ことを特徴とする請求項17記載の記録媒体。

19. 前記ムービーモードのプログラムは、ボタンコマンドを含み

25 ボタンコマンドは、エンハンスドモードのプログラムへと分岐するコマンドであり、

ボタンコマンドは、動画データ、副映像データと多重化されて多重化ストリームとして記録媒体に記録され、

副映像データは、ボタンの画像データであり、前記ボタンコマンドは、30 ボタンの画像データに対して確定操作がなされた際、実行される

ことを特徴とする請求項 1 3 記載の記録媒体。

20. 前記エンハンスモードとは、

ブラウザに、プログラムを実行させるモードであり、

エンハンスドモードプログラムは、マークアップ言語で記述される

5 ことを特徴とする請求項 1 3 記載の記録媒体。

21. 前記ムービーモードのプログラムは、ボタンコマンドを含み

ボタンコマンドは、エンハンスドモードのプログラムへと分岐するコ
マンドであり、

ボタンコマンドは、動画データ、副映像データと多重化されて多重化

10 ストリームとして記録媒体に記録され、

副映像データは、ボタンの画像データであり、前記ボタンコマンドは、
ボタンの画像データに対して確定操作がなされた際、実行される
ことを特徴とする請求項 2 0 記載の記録媒体。

22. 前記モードには、仮想マシンに、プログラムを実行させる仮想

15 マシンモードと、

ブラウザに、プログラムを実行させるブラウザモードとがあり、

前記分岐とは、

仮想マシンモードのプログラム、ブラウザモードのプログラム間の分
岐である

20 ことを特徴とする請求項 1 記載の記録媒体。

23. 動画データ、複数のプログラム、各プログラムについてのイン
デックスからなるテーブルが記録された記録媒体についての再生装置で
あって、

ムービーモードに対応したモジュールと、エンハンスドモードに対応

25 したモジュールとを含み、記録媒体に記録されたプログラムのそれぞれ
を実行する複数のモジュールと、

プログラム間の分岐を制御するマネージャとを備え、

前記テーブルは、各プログラムがムービーモード及びエンハンスドモ
ードの何れに属するかを示す情報であり、

30 前記分岐は、プログラムにおいてテーブルを介した間接参照の形式で

記述され、

前記マネージャは、

プログラムの分岐にあたって、テーブルを参照して、どのモジュールに、分岐先プログラムを実行させるかを決定する

5 ことを特徴とする再生装置。

24. 前記テーブルは、複数プログラムのそれぞれに対応した複数のインデックスを含み、

各インデックスは、

対応するプログラムがエンハンスドモードに属するか、ムービーモードに属するかを示しており、

前記分岐コマンドにおける間接参照の形式とは、

インデックスについてのラベルを用いて、分岐先プログラムを指定することである

ことを特徴とする請求項23記載の再生装置。

15 25. 前記マネージャによる実行モジュールの決定は、

分岐先プログラムを指定するラベルが、どのインデックスに割り当てられているかを識別し、そのインデックスが、エンハンスドモードに属するか、ムービーモードに属するかを判定することにより行われることを特徴とする請求項24記載の再生装置。

26. 前記エンハンスドモードモジュールによるプログラム実行が不可能である場合、前記マネージャは、

分岐先プログラムを指定するラベルが、エンハンスドモードに属しているインデックスに割り当てられたとしても、代わりのムービーモードのプログラムを、ムービーモードモジュールに実行させ、

前記テーブルにおける予備のインデックスを有し、

前記代わりのムービーモードプログラムは、この予備のインデックスに対応していることを特徴とする請求項25記載の再生装置。

27. 前記再生装置はレジスタと、

各モードのモジュールからの関数呼出しに応じて、機能を実行する再生

30 制御エンジンとを備え、

再生制御エンジンが実行する機能とは、

モジュールから値を受け取ってレジスタに設定する機能(1)、

レジスタから値を取得してモジュールに引き渡す機能(2)、

予め定められた再生経路に従い、動画データを再生する機能(3)の何

5 れかである

ことを特徴とする請求項23記載の再生装置。

28. 前記再生装置はレジスタと、再生制御エンジンとを備え、前記分岐とは、ムービーモードプログラムからエンハンスドモードプログラムへの分岐であり、

10 ムービーモードプログラムによる再生制御手順とは、再生装置のレジスタへの値の設定を再生制御エンジンに行わせることであり、

エンハンスドモードプログラムによる再生制御手順とは、レジスタに設定された値を、再生制御エンジンに読み出させることである

ことを特徴とする請求項27記載の再生装置。

15 29. 前記レジスタ設定値とは、

再生装置における音声設定(1)、再生装置における副映像設定(2)、再生装置におけるアングル設定(3)、現在再生されているタイトル(4)、現在再生されているチャプター(5)、現在の再生時点(6)の何れかを示す値である

20 ことを特徴とする請求項28記載の再生装置。

30. 前記記録媒体にはプレイリスト情報が記録されており、

プレイリスト情報は、動画データにおける再生区間を示す情報を、再生順序に従って配列することにより再生経路を定義する情報であり、

再生制御エンジンによる再生制御は、

25 プレイリスト情報に示される再生経路に基づく

ことを特徴とする請求項28記載の再生装置。

31. 前記プログラムは、再生制御エンジンが実行する再生機能を呼び出すための関数呼出を含み、

この関数呼出には、2つの引数があり、

30 そのうち1つ目の引数は、プレイリスト情報を指定するものであり、

2つ目の引数は、再生経路における開始点を指定するものであることを特徴とする請求項30記載の再生装置。

32. 前記再生手順における開始点指定は、

再生区間、再生時刻、チャプターの何れかを用いて行われる

5 ことを特徴とする請求項31記載の再生装置。

33. 前記記録媒体においてプレイリスト情報は、フラグが付加され
て記録され、

フラグは、本プレイリスト情報を用いた動画データの再生を、ムービー^{モード}で許可するか、エンハンスドモードで許可するかを示し、

10 再生制御エンジンは、フラグに示されるモードの実行時において、プレイリスト情報に基づく再生制御を行う

ことを特徴とする請求項31記載の再生装置。

34. 前記分岐とは、ムービーモードプログラムからエンハンスドモードプログラムへの分岐であり、

15 ムービーモードプログラムによる動画データの再生手順とは、

プレイリスト情報により示される再生経路における開始点を指定して、再生制御エンジンに再生を行わせることであり、

エンハンスドモードプログラムによる動画データの再生手順とは、

同じプレイリスト情報により示される再生経路における開始点を指定

20 して、再生制御エンジンに再生を行わせることである

ことを特徴とする請求項31記載の再生装置。

35. 前記エンハンスモードに対応するモジュールは、仮想マシンを備え、

エンハンスドモードのプログラムは、仮想マシン向けプログラミング

25 言語により記述される

ことを特徴とする請求項23記載の再生装置。

36. 前記再生制御エンジンは、動画データの再生に同期したイベントの発生を行い、

エンハンスドモードプログラムは、イベントハンドラを含み、

30 前記モジュールは、再生制御エンジンがイベントを発生すると、イベ

ントハンドラを実行する

ことを特徴とする請求項 3 5 記載の再生装置。

37. 前記イベントとは、

現在の再生位置が、動画データの再生時間軸における所定の時点に到
5 達したこととを示すイベント(i)、

動画データの再生時間軸において、所定の時間間隔だけ、現在の再生
位置が進行したことを示すイベント(ii)の何れかである

ことを特徴とする請求項 3 6 記載の再生装置。

38. 前記記録媒体はマーク情報が記録されており、

10 マーク情報は、

前記再生時間軸における所定の時点、及び、所定の時間間隔を定義す
る情報であり、

再生制御エンジンによるイベント発生は、このマーク情報に基づいて
行われる

15 ことを特徴とする請求項 3 7 記載の再生装置。

39. 前記再生装置は、ユーザ操作を受け付ける受付手段を備え、

前記イベントとは、

動画データの再生時間軸上の所定の時間帯において、受付手段がユー
ザ操作を受け付けたことを示すイベントである

20 ことを特徴とする請求項 3 6 記載の再生装置。

40. 前記受付手段が受け付ける操作とは、

動画データの再生時において、早送り、巻戻し、ポーズの操作である
ことを特徴とする請求項 3 9 記載の再生装置。

41. 前記ムービーモードのプログラムは、ボタンコマンドを含み
25 ボタンコマンドは、エンハンスドモードのプログラムへと分岐するコ
マンドであり、

ボタンコマンドは、動画データ、副映像データと多重化されて多重化
ストリームとして記録媒体に記録され、

副映像データは、ボタンの画像データであり、

30 前記再生装置は

多重化ストリームを分離して、ボタンコマンド、動画データ、副映像データを得るデマルチプレクサと、

ボタンの画像データをデコードするイメージデコーダと、

動画データをデコードするビデオデコーダとを備え、

5 前記モジュールは、ボタンの画像データに対して確定操作がなされた際、ボタンコマンドを実行する

ことを特徴とする請求項35記載の再生装置。

42. 前記エンハンスマードとは、

ブラウザに、プログラムを実行させるモードであり、

10 エンハンスマードプログラムは、マークアップ言語で記述されることを特徴とする請求項23記載の再生装置。

43. 前記ムービーモードのプログラムは、ボタンコマンドを含み
ボタンコマンドは、エンハンスマードのプログラムへと分岐するコ
マンドであり、

15 副映像データは、ボタンの画像データであり、

ボタンコマンドは、動画データ、副映像データと多重化されて多重化
ストリームとして記録媒体に記録され、

前記再生装置は

多重化ストリームを分離して、ボタンコマンド、動画データ、副映像
20 データを得るデマルチプレクサと、

ボタンの画像データをデコードするイメージデコーダと、

動画データをデコードするビデオデコーダとを備え、

前記モジュールは、ボタンの画像データに対して確定操作がなされた
際、ボタンコマンドを実行する

25 ことを特徴とする請求項42記載の再生装置。

44. 前記モードには、仮想マシンに、プログラムを実行させる仮想
マシンモードと、

ブラウザに、プログラムを実行させるブラウザモードとがあり、

前記分岐とは、

30 仮想マシンモードのプログラム、ブラウザモードのプログラム間の分

岐である

ことを特徴とする請求項23記載の再生装置。

45. 動画データ、複数のプログラム、各プログラムについてのインデックスからなるテーブルが記録された記録媒体についての再生処理プログラムであって、
5 ムービーモードに対応した実行ステップと、エンハンスドモードに対

応した実行ステップとを含み、各実行ステップは、記録媒体に記録されたプログラムのそれぞれを実行する複数との実行ステップと、

プログラム間の分岐を制御する制御ステップとをコンピュータに行わ
10 せ、

前記テーブルは、各プログラムがムービーモード及びエンハンスドモードの何れに属するかを示す情報であり、

前記分岐は、プログラムにおいてテーブルを介した間接参照の形式で記述され、

15 前記制御ステップは、

プログラムの分岐にあたって、テーブルを参照して、どの実行ステップに、分岐先プログラムを実行させるかを決定する
ことを特徴とする再生処理プログラム。

46. 動画データ、複数のプログラム、各プログラムについてのインデックスからなるテーブルが記録された記録媒体についての再生方法であって、
20

ムービーモードに対応した実行ステップと、エンハンスドモードに対応した実行ステップとを含み、各実行ステップは、記録媒体に記録されたプログラムのそれぞれを実行する複数の実行ステップと、

25 プログラム間の分岐を制御する制御ステップとを含み、

前記テーブルは、各プログラムがムービーモード及びエンハンスドモードの何れに属するかを示す情報であり、

前記分岐は、プログラムにおいてテーブルを介した間接参照の形式で記述され、

30 前記制御ステップは、

プログラムの分岐にあたって、テーブルを参照して、どの実行ステップに、分岐先プログラムを実行させるかを決定することを特徴とする再生方法。

4 7. 記録媒体の記録方法であって、
5 アプリケーションデータを作成するステップと、
作成したデータを記録媒体に記録するステップとを有し、
前記アプリケーションデータは、動画データ、複数のプログラム、テ
ーブルを含み、
各プログラムは、
10 動画データの再生制御手順を示し、
テーブルは、記録媒体における各プログラムの識別情報(1)、各プロ
グラムがムービーモード及びエンハンスドモードの何れに属するかを示
す情報(2)を含んでおり、
何れか1つのプログラムは、分岐コマンドを含み、
15 前記分岐コマンドは、テーブルを介した間接参照の形式で、分岐先を
指定している
ことを特徴とする記録方法。

図1

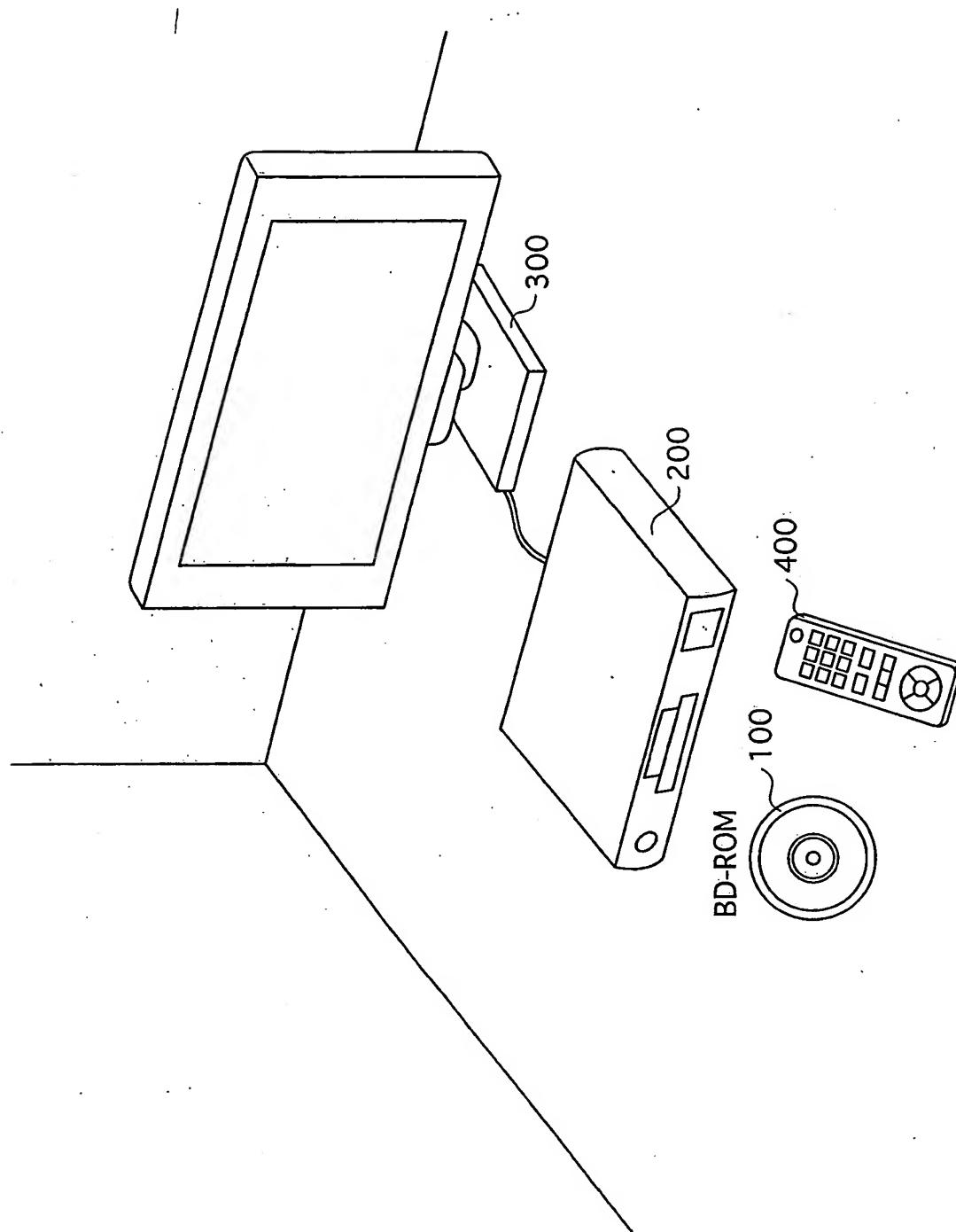


図2

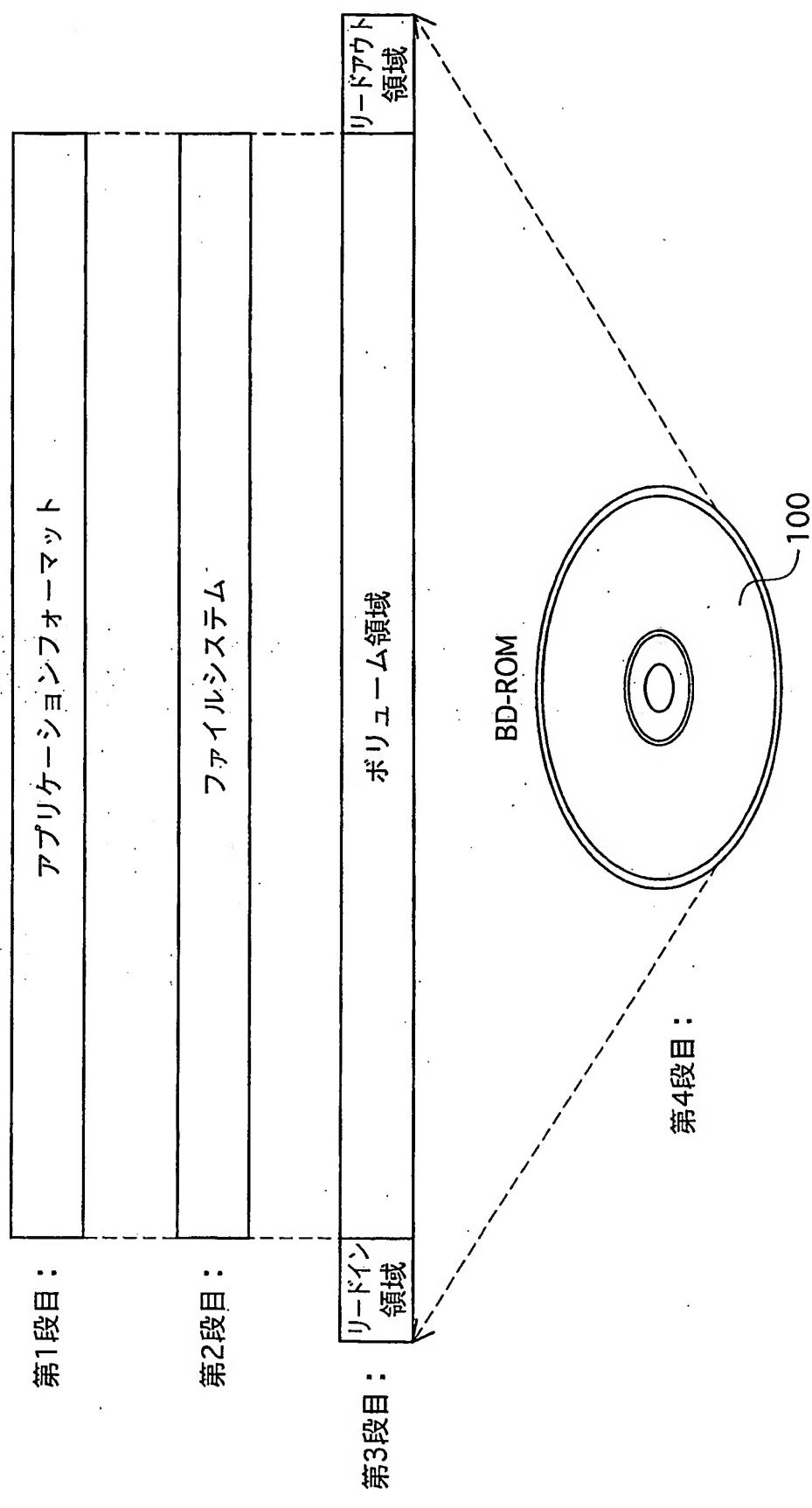


図3

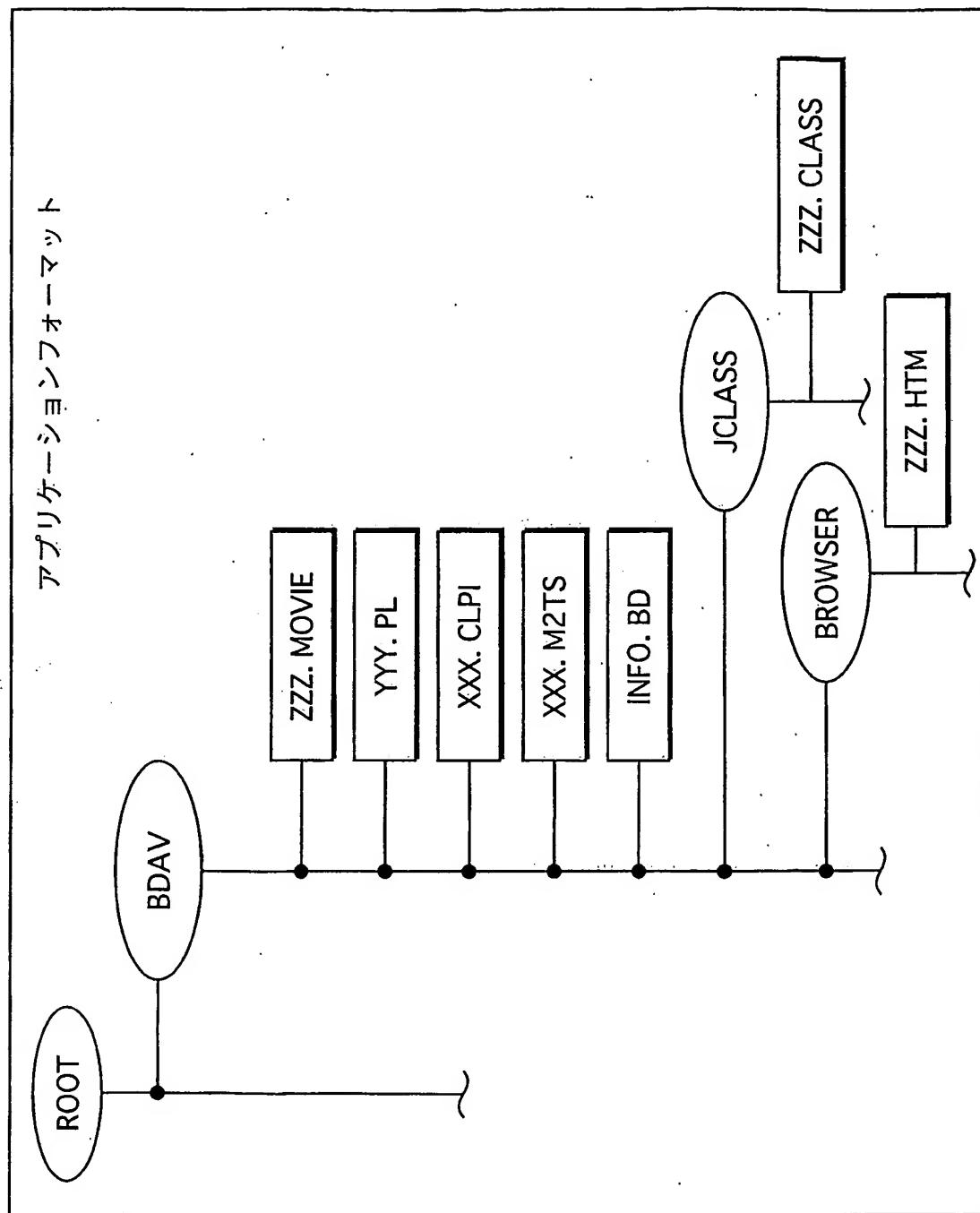


図4

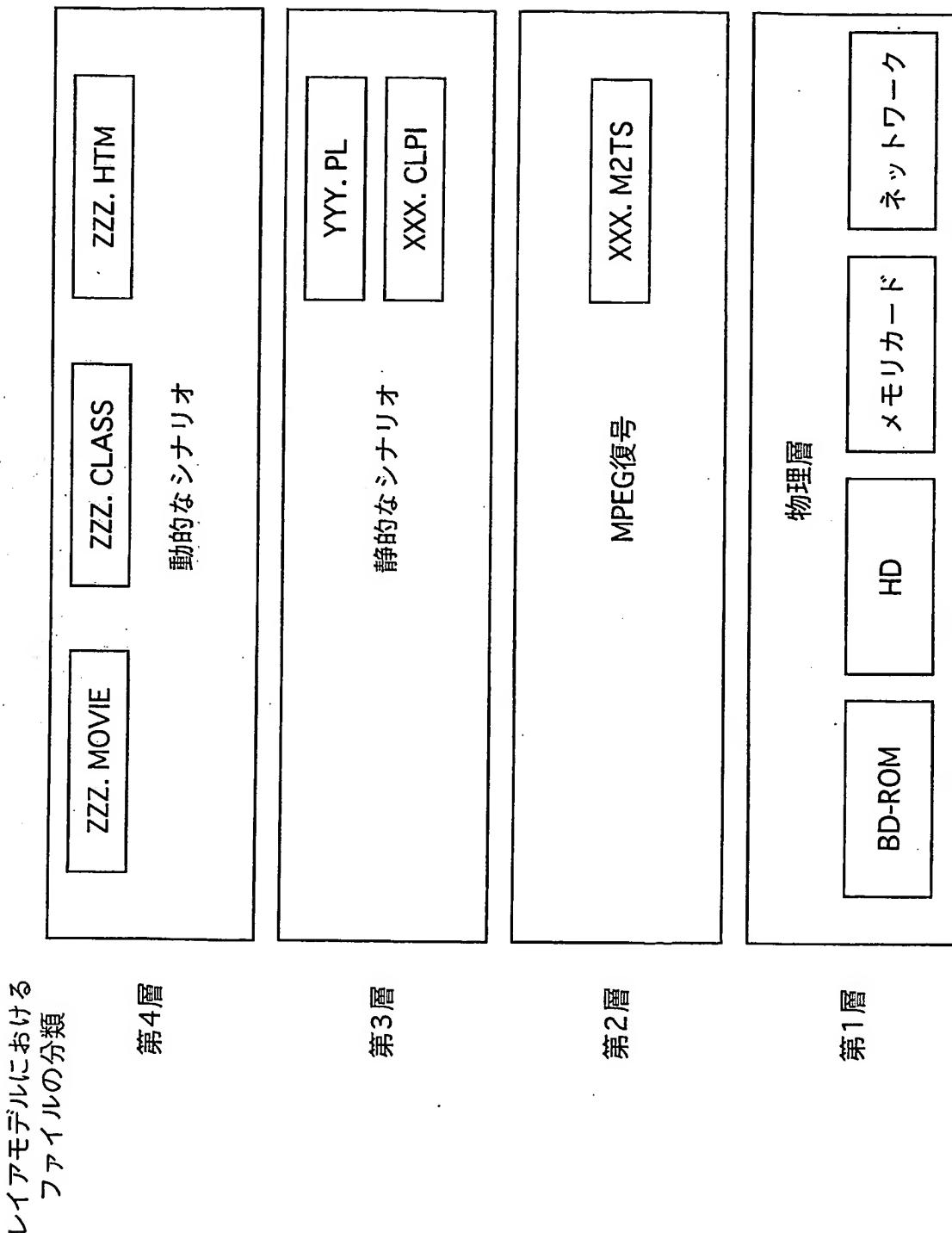


図5

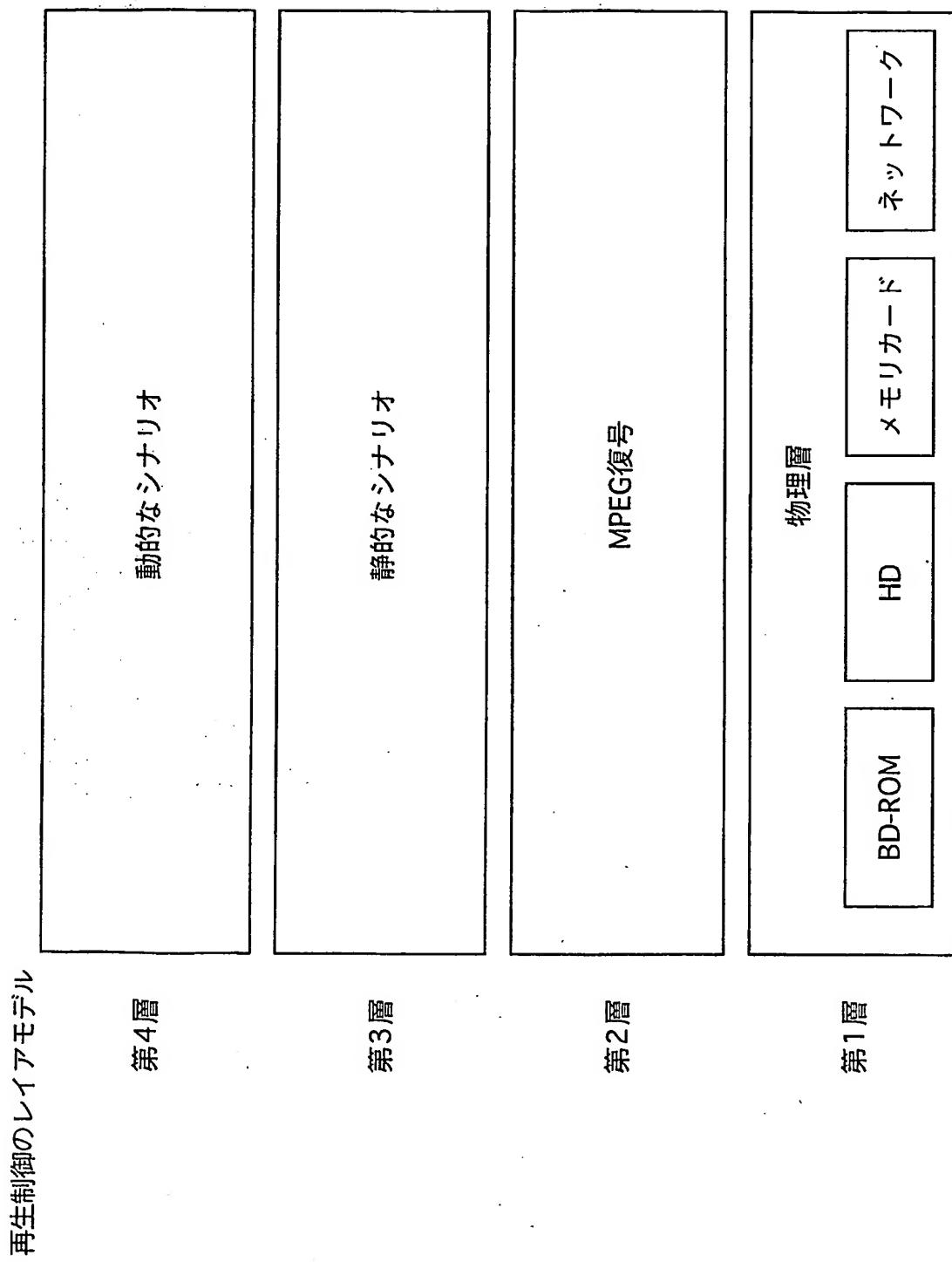


図6

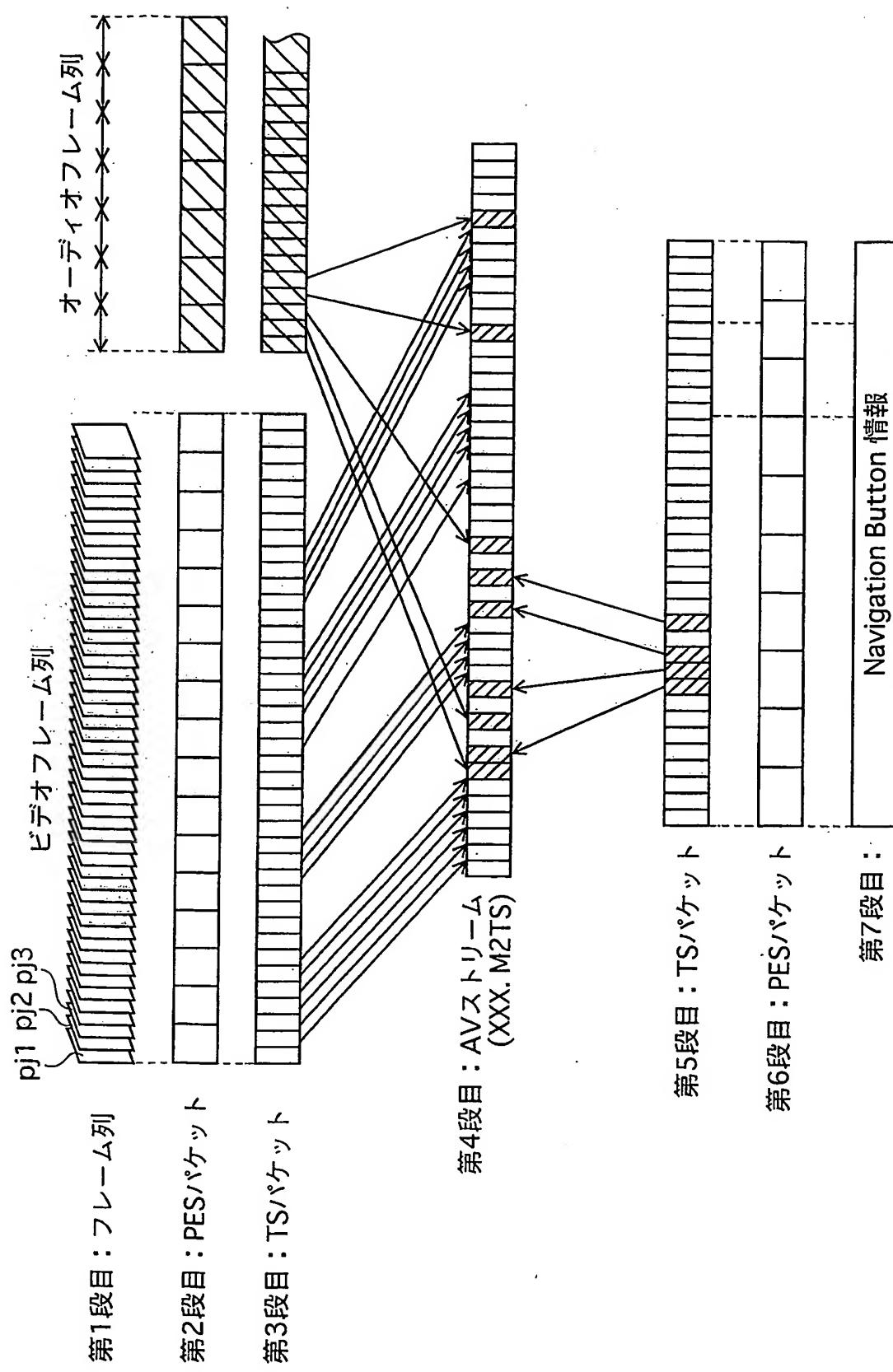


図7

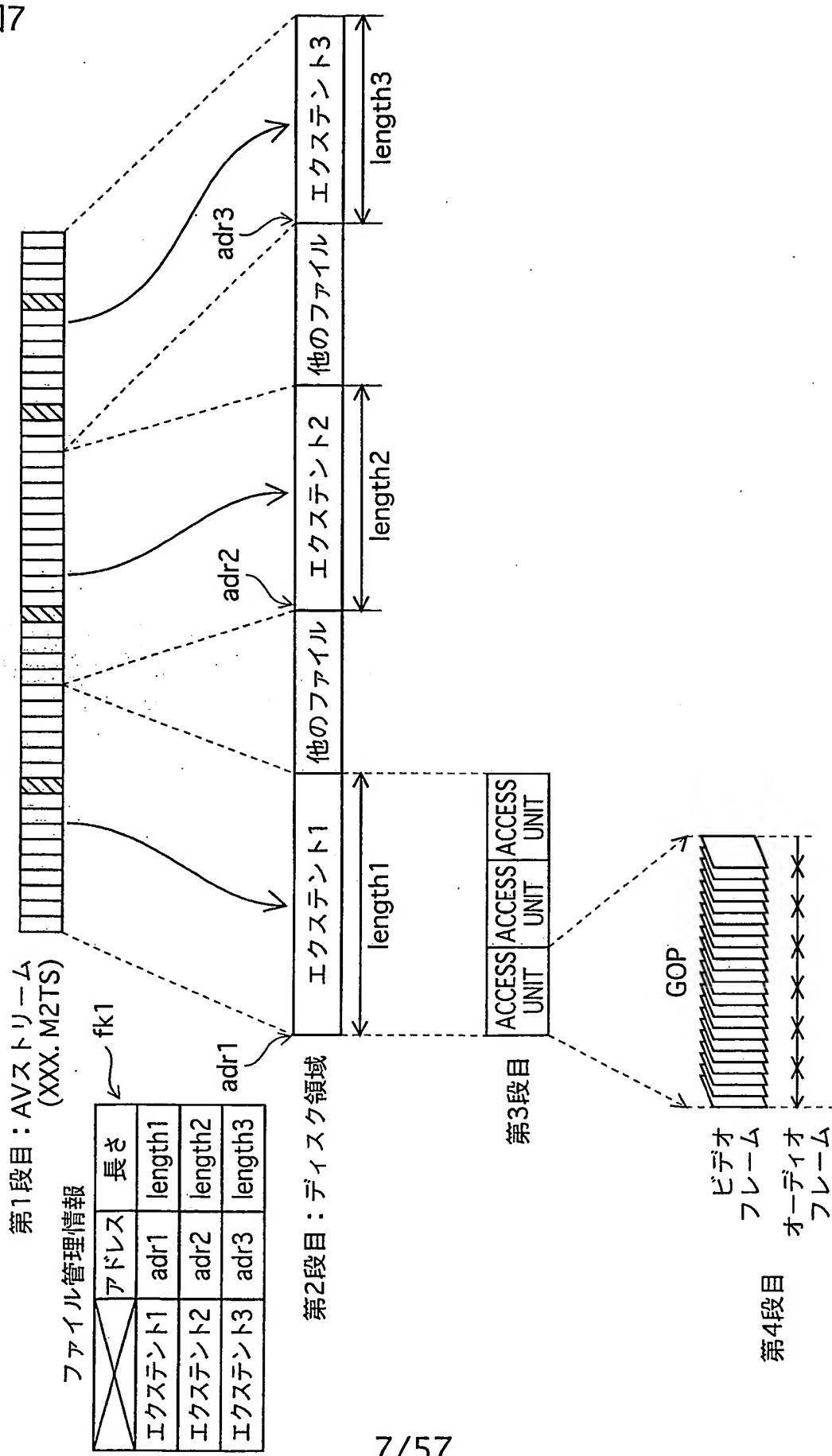


図8

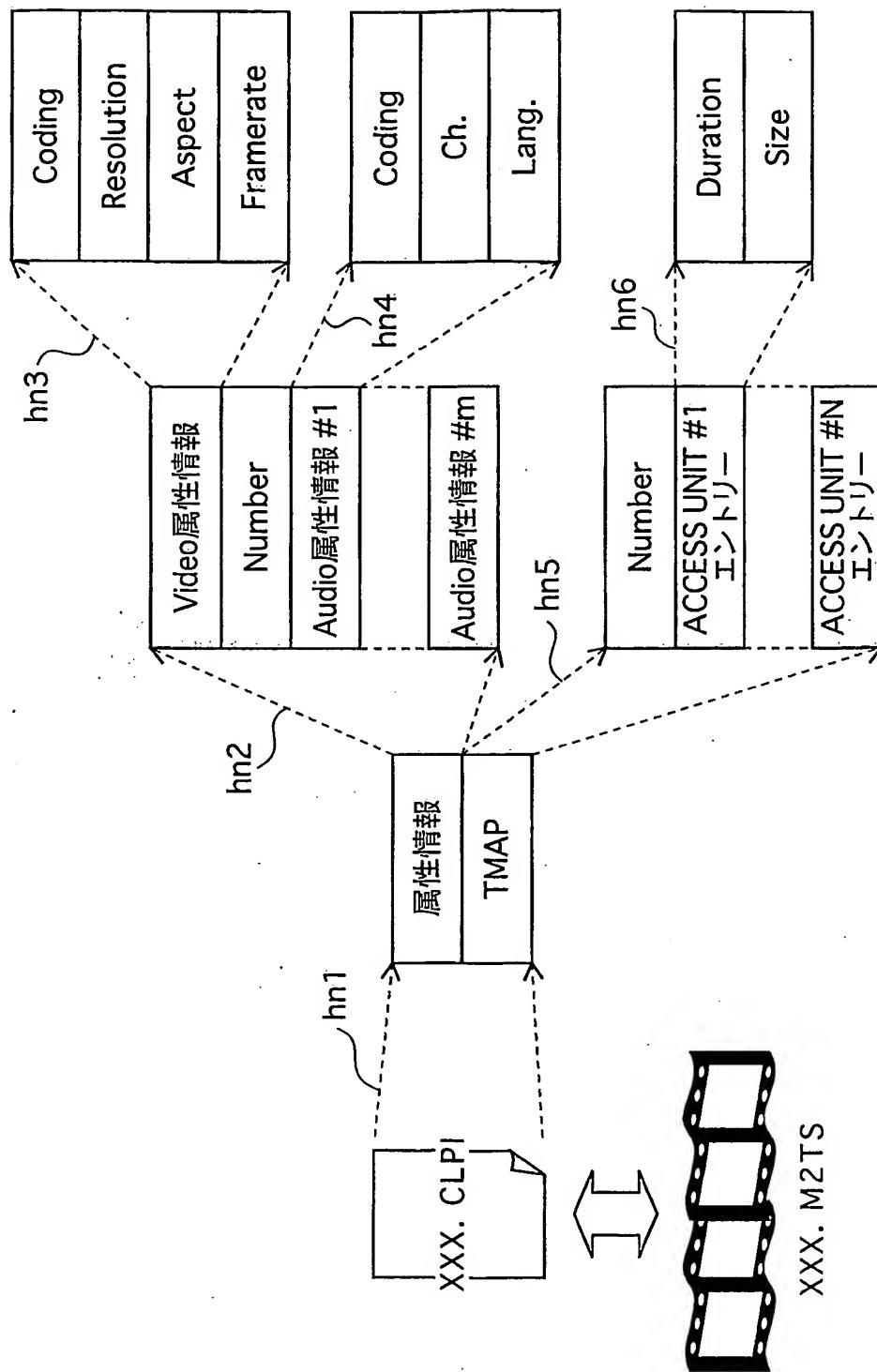


図9

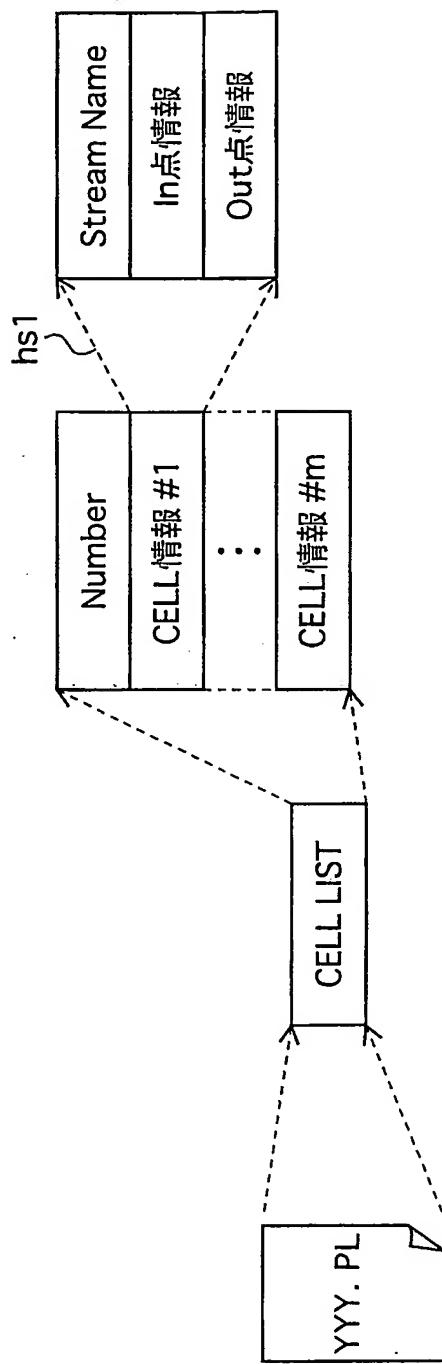


図10

静的なシナリオ

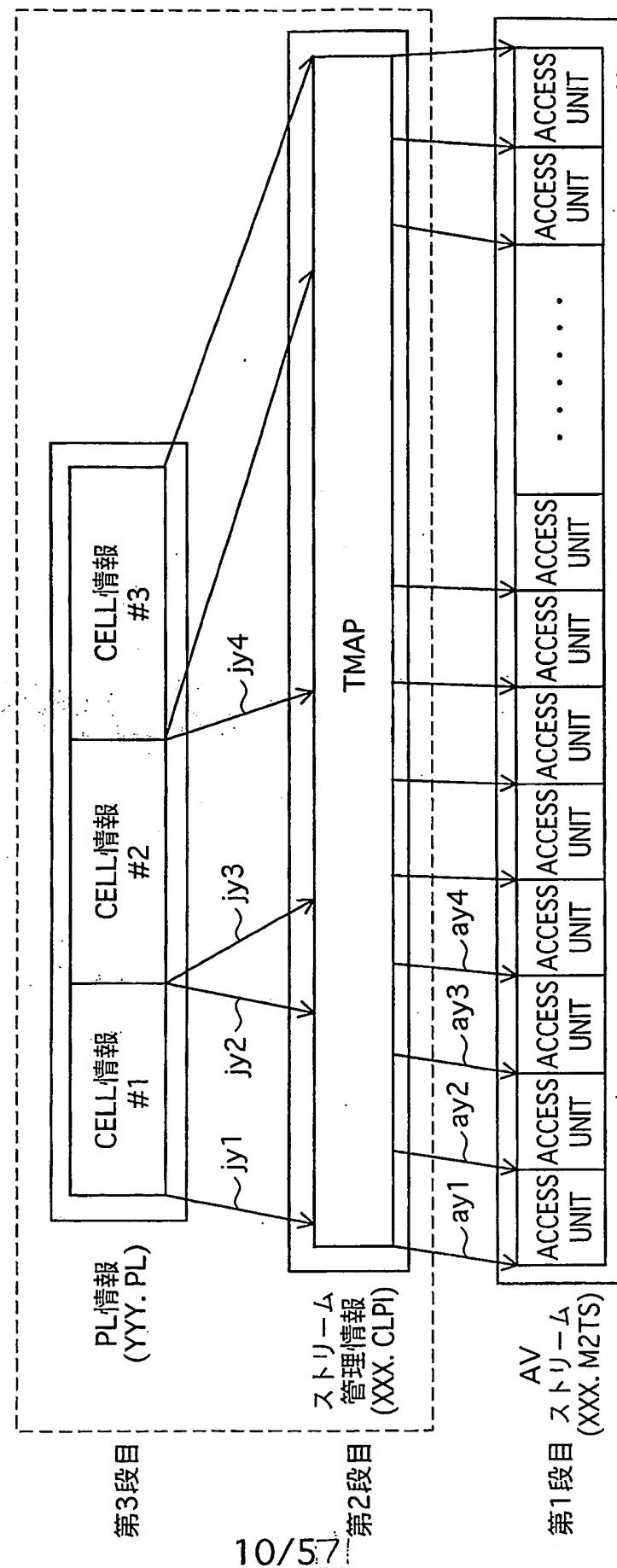


図11

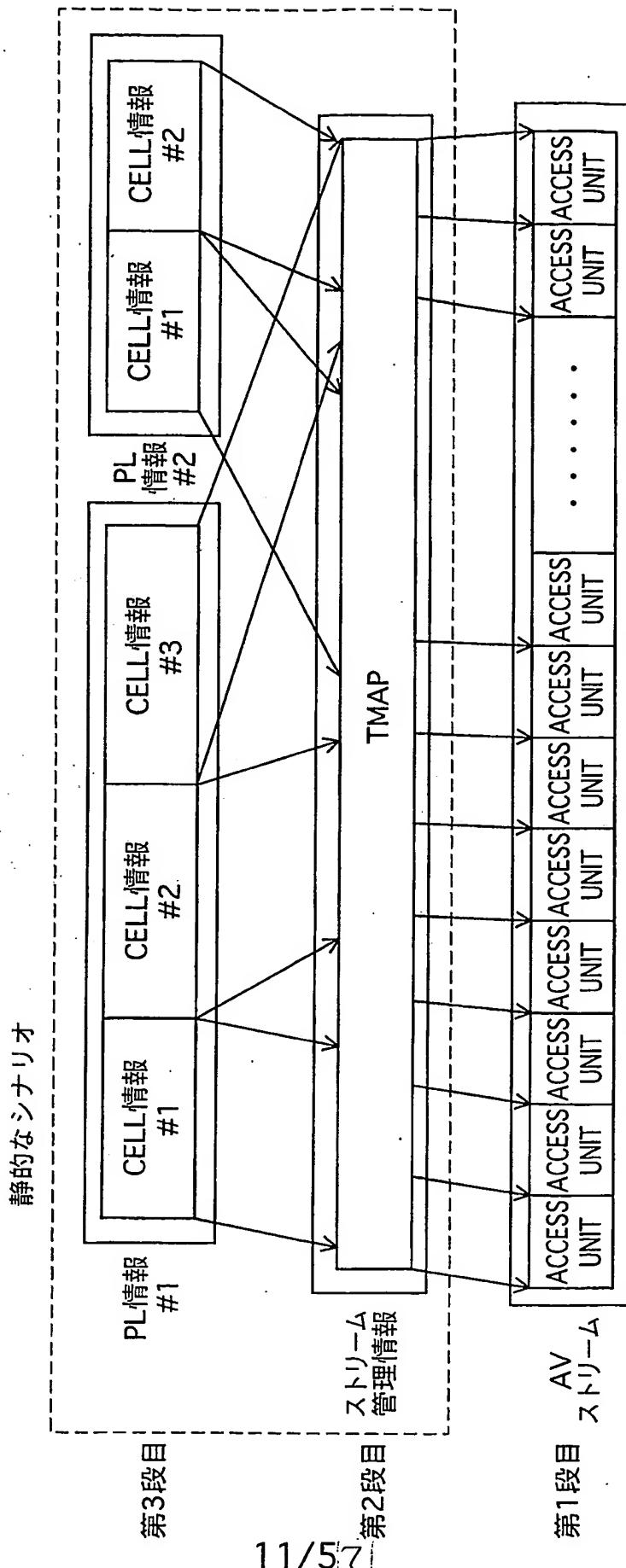


図12

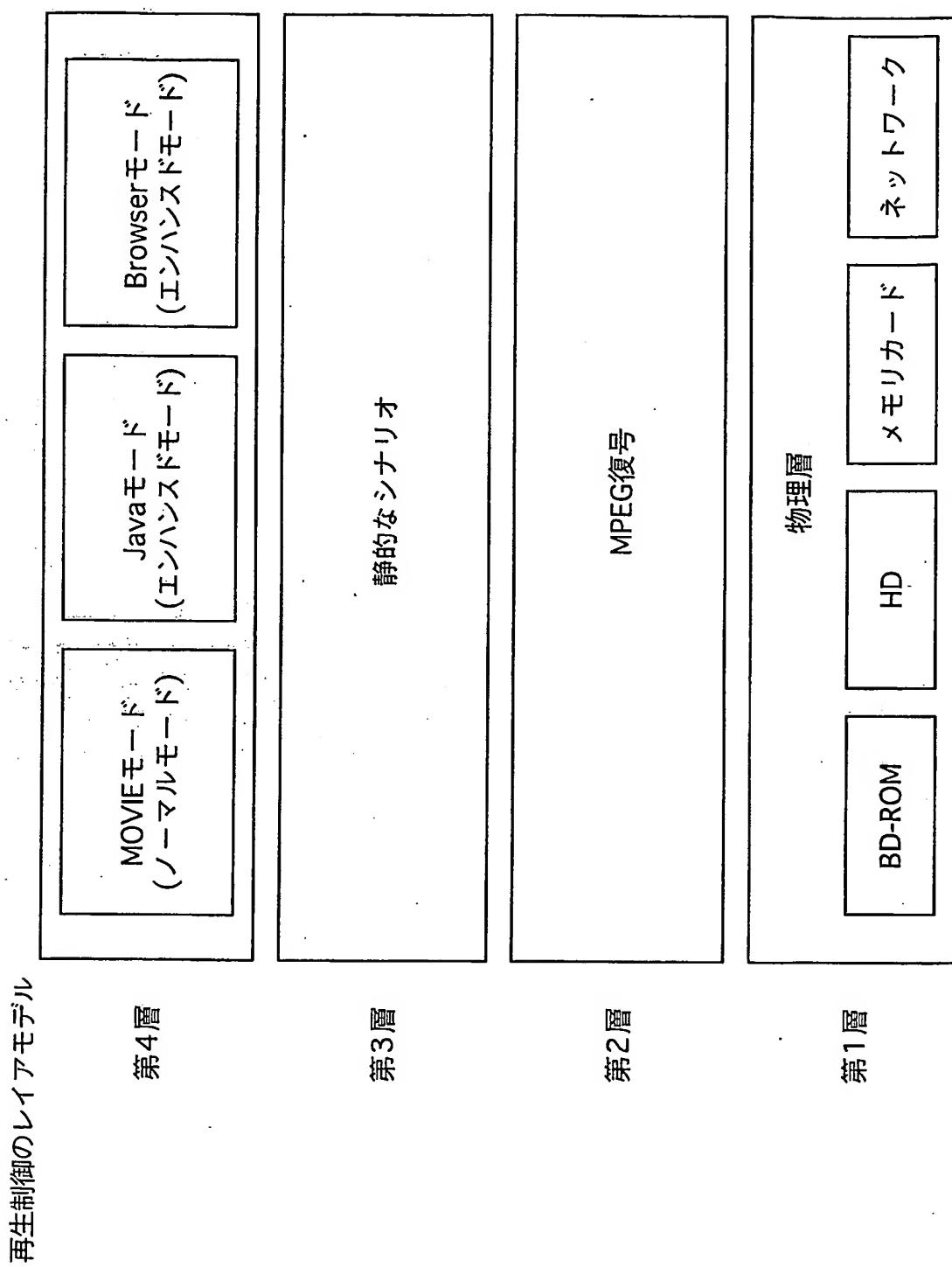


図13

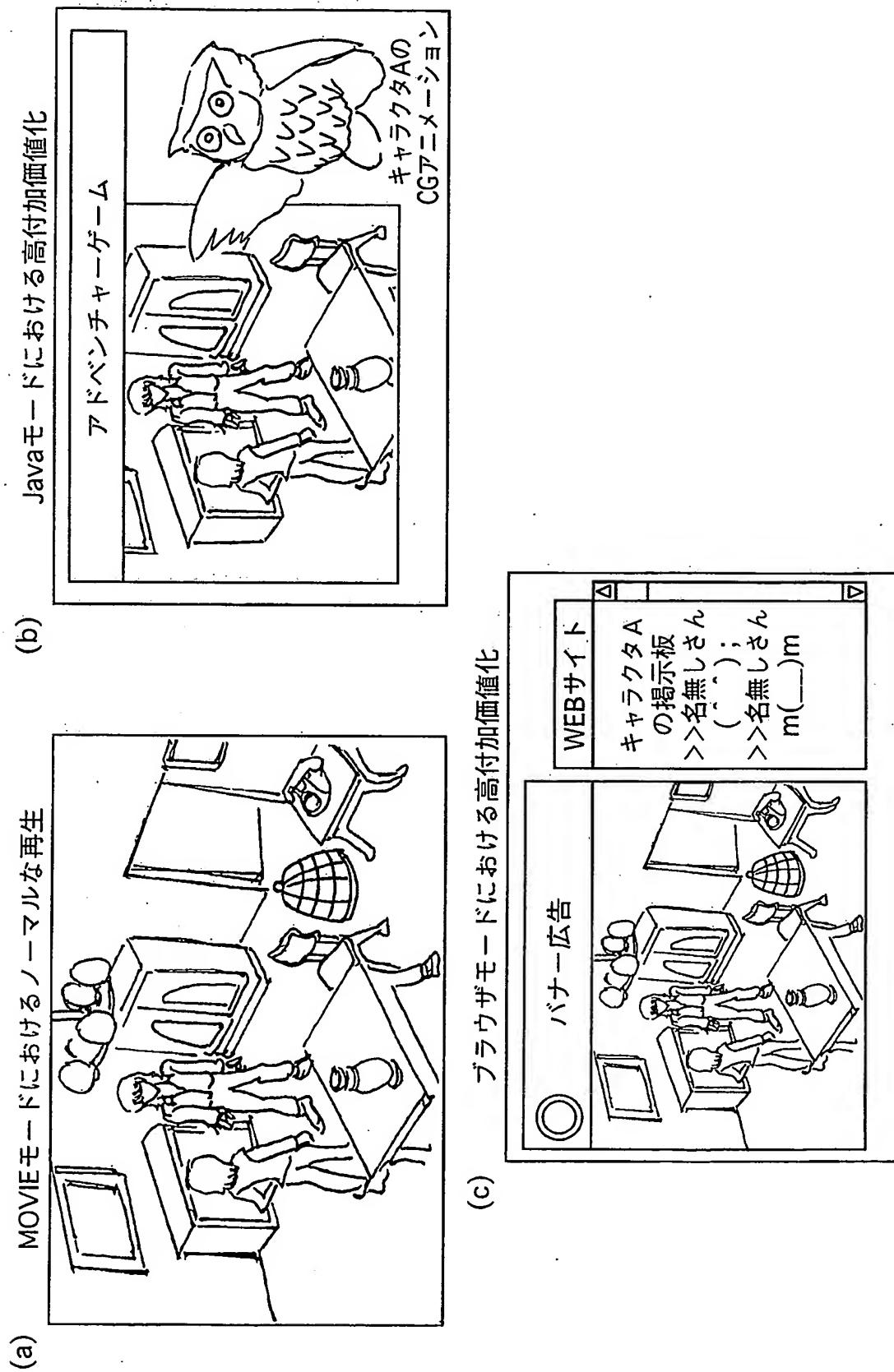


図14

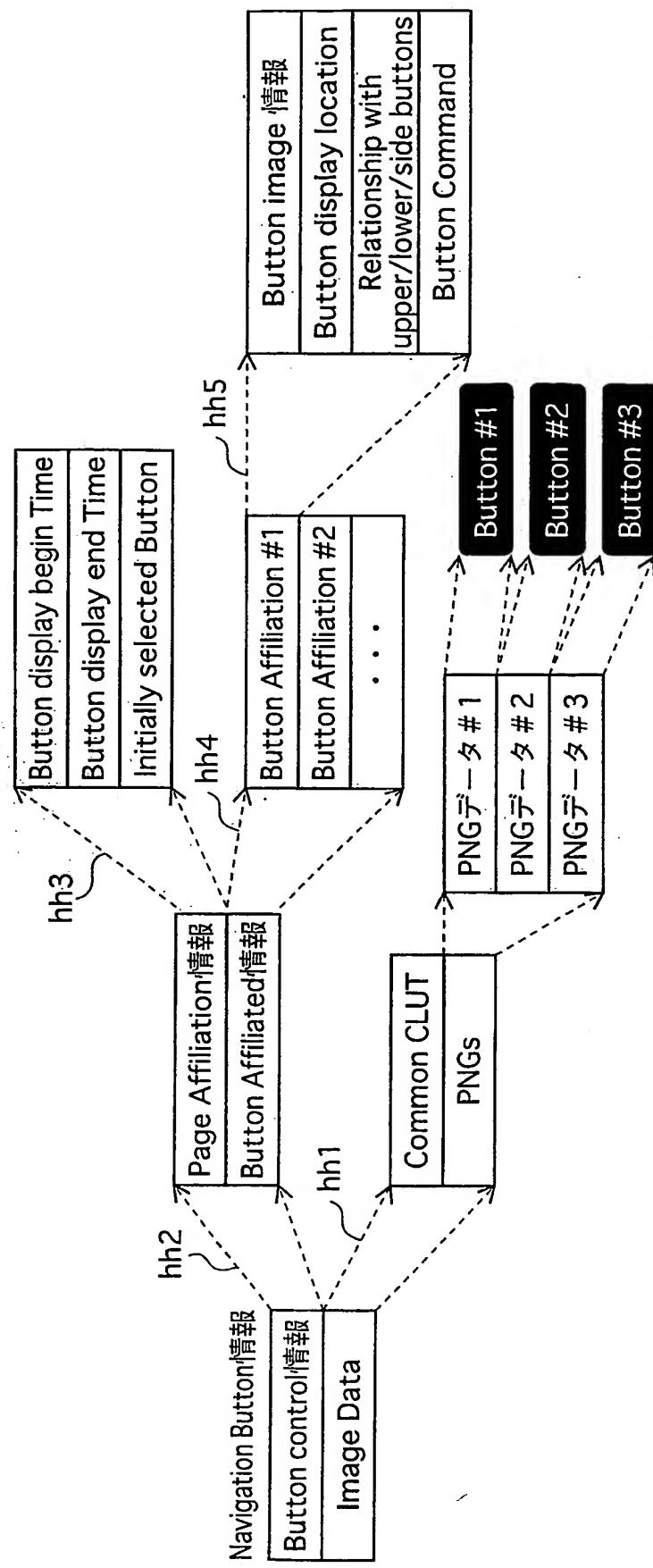


図15

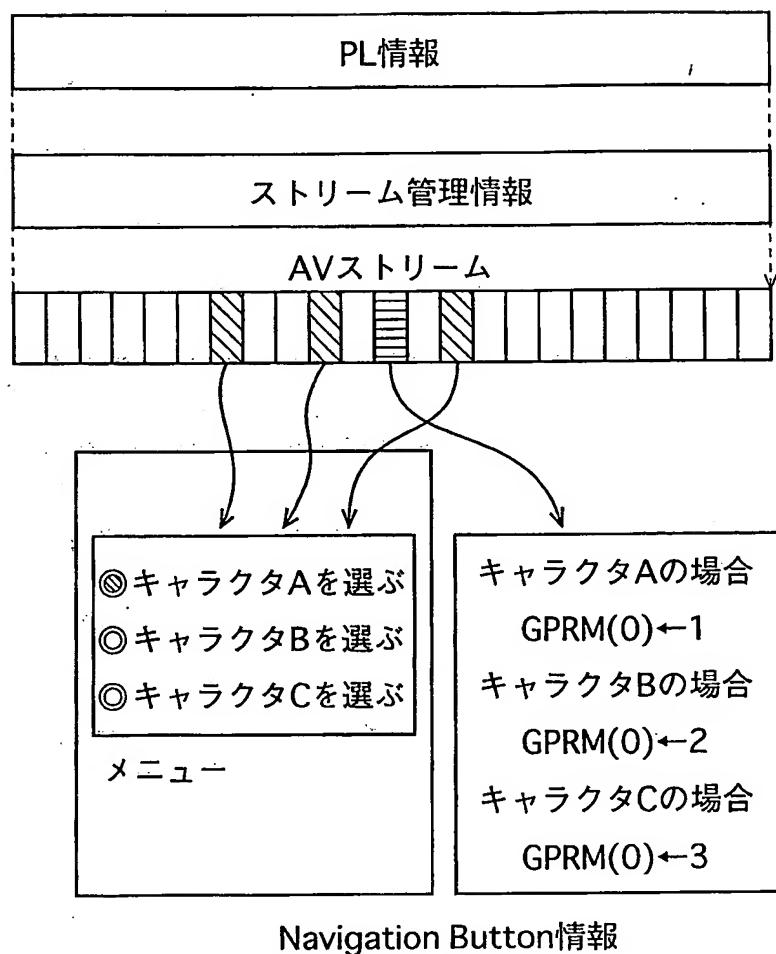


図16

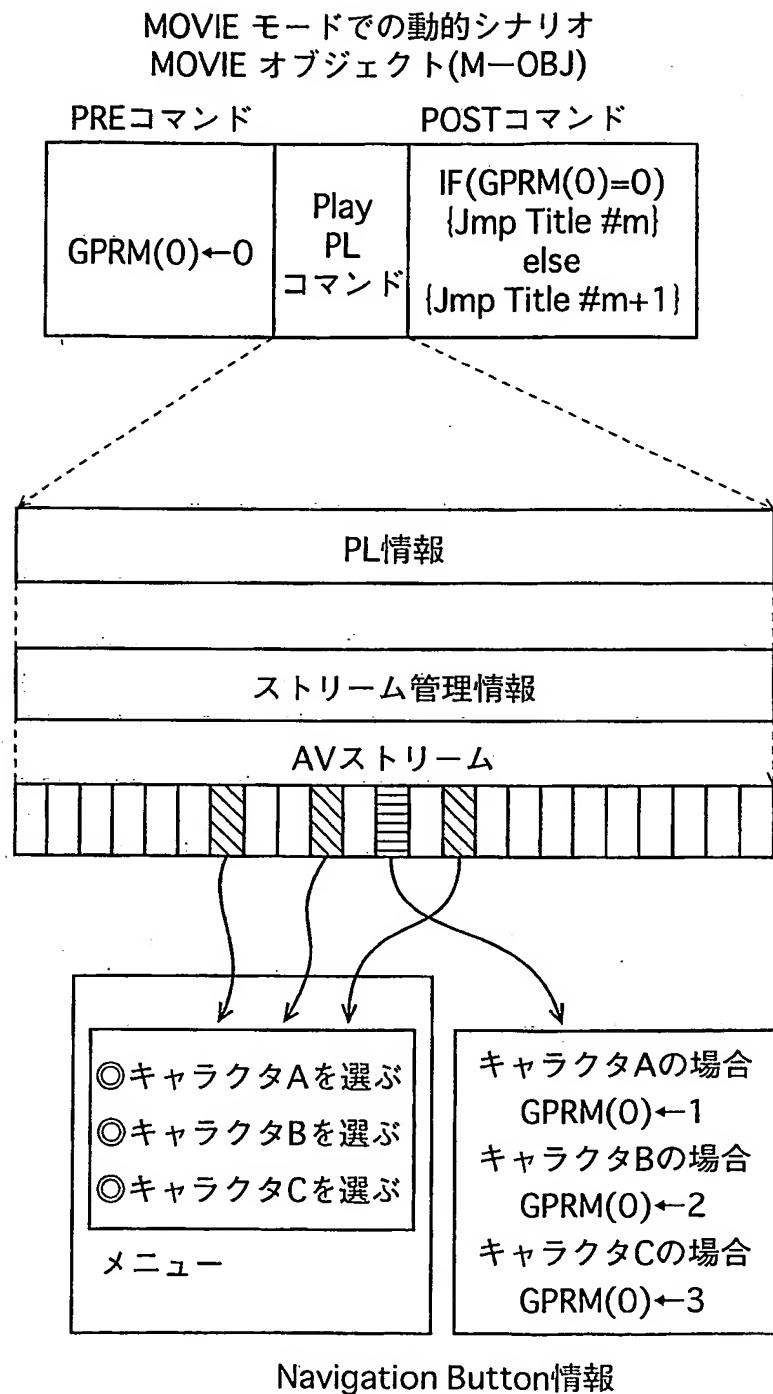


図17

Javaオブジェクト(J-OBJ)

```
IF(GPRM(0)==1)
{ A.draw Character ();PlayPL(PL#1,CELL#2);}
else { if (GPRM(0)==2)
{ B.draw Character();PlayPL(PL#2,CELL#1);}}
else { if (GPRM(0)==3)
{ C.draw Character();PlayPL(PL#1,CELL#3);}}}
```

ZZZ.CLASS

図18

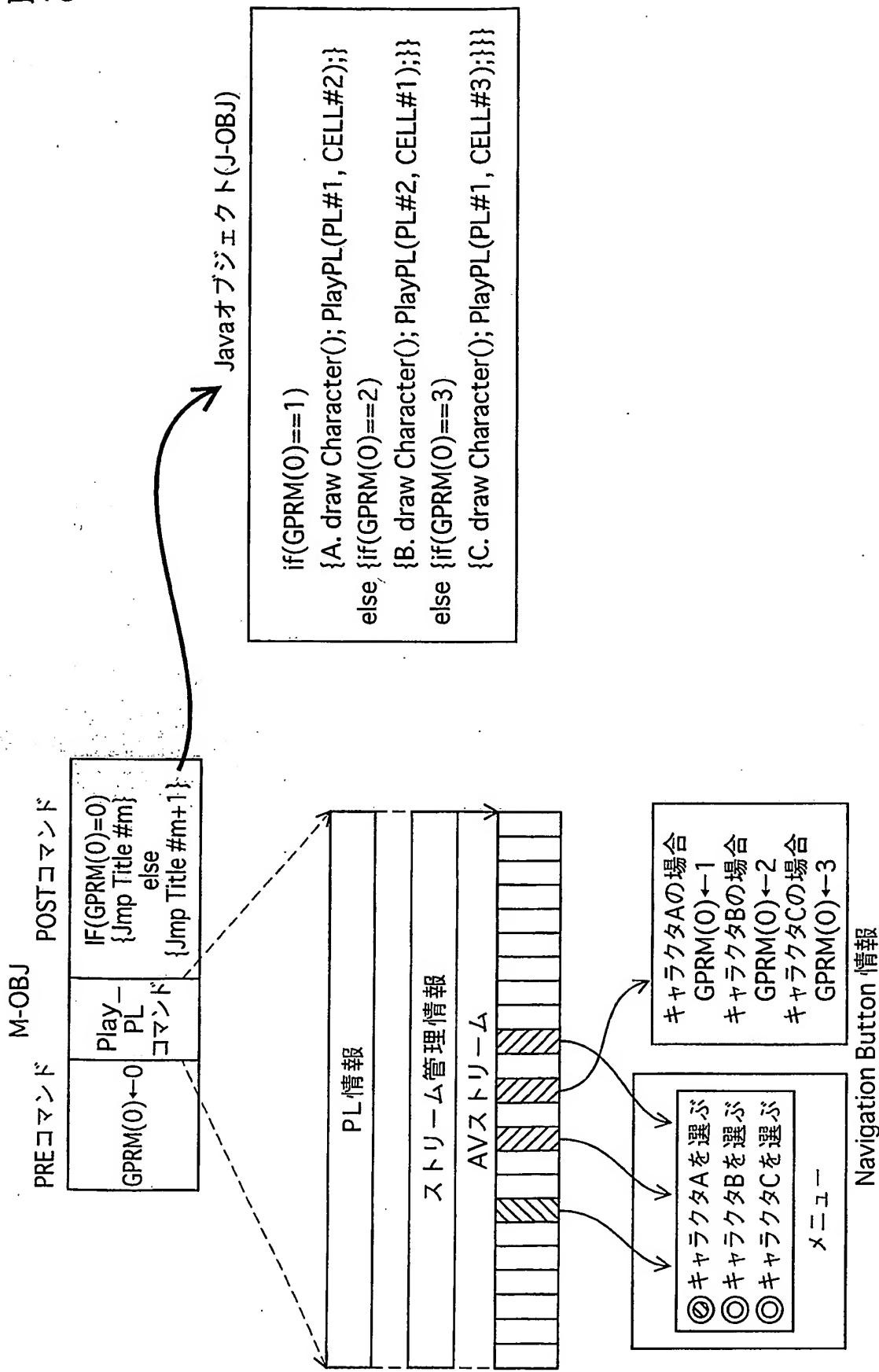


図19

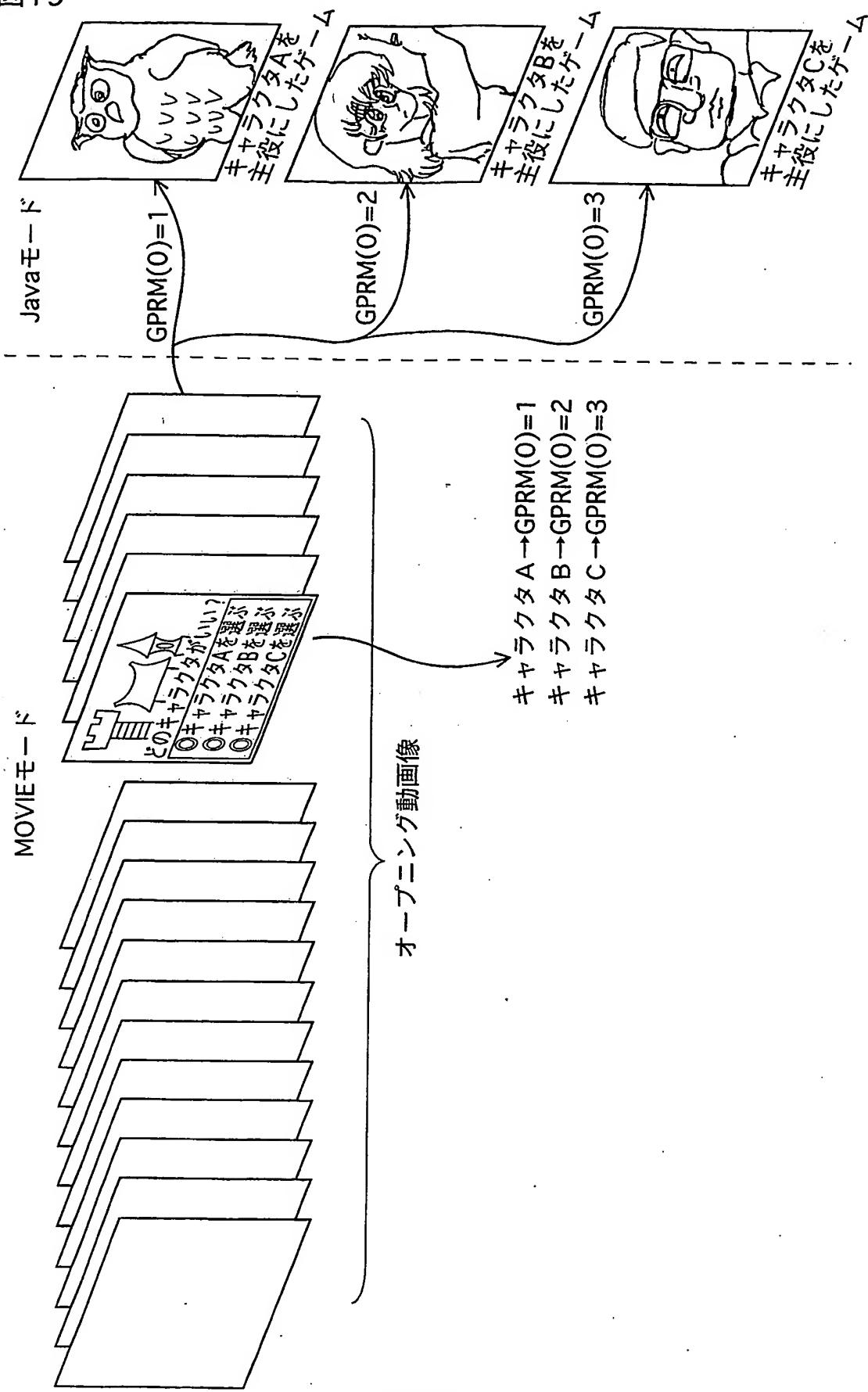


図20

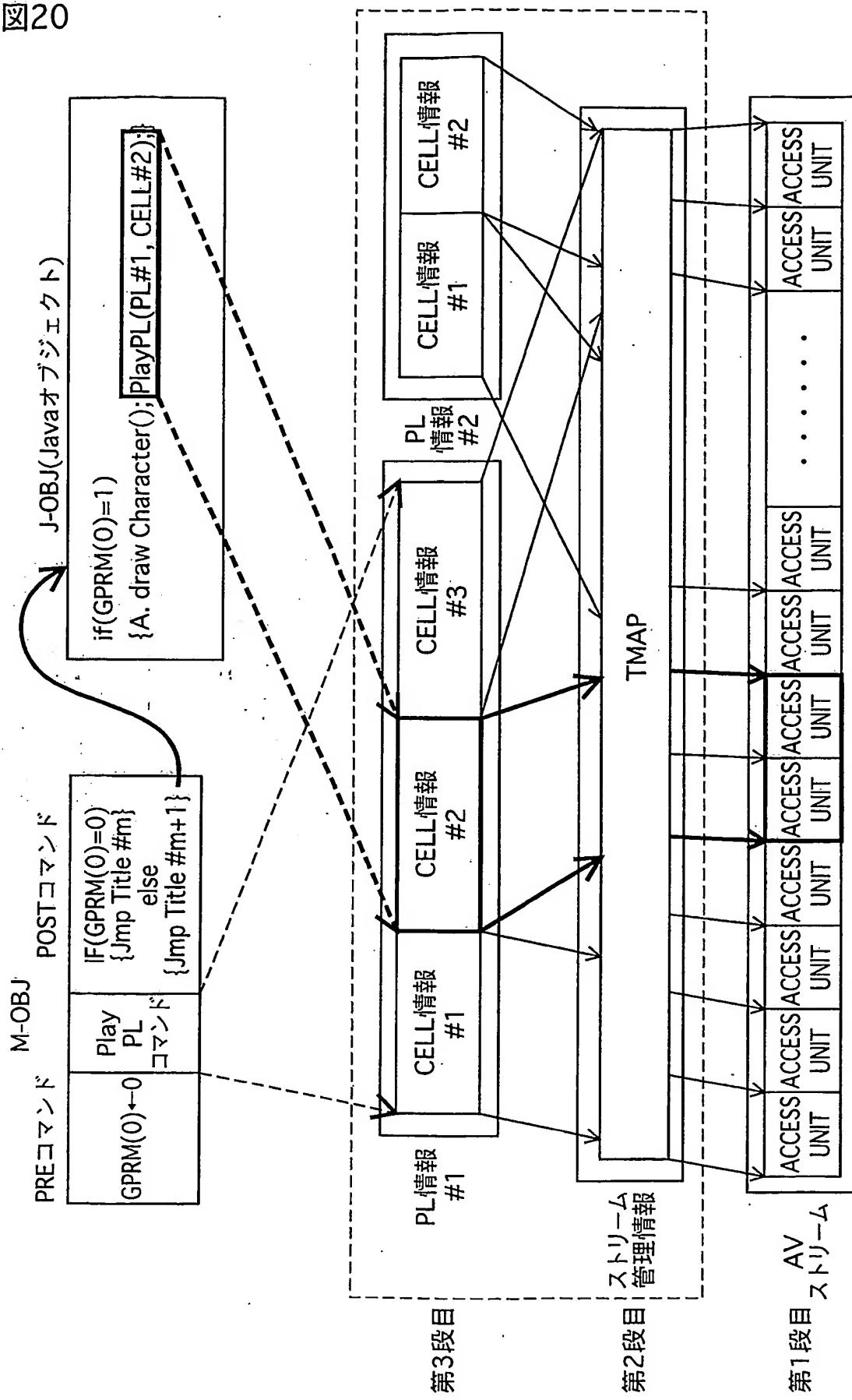


図21

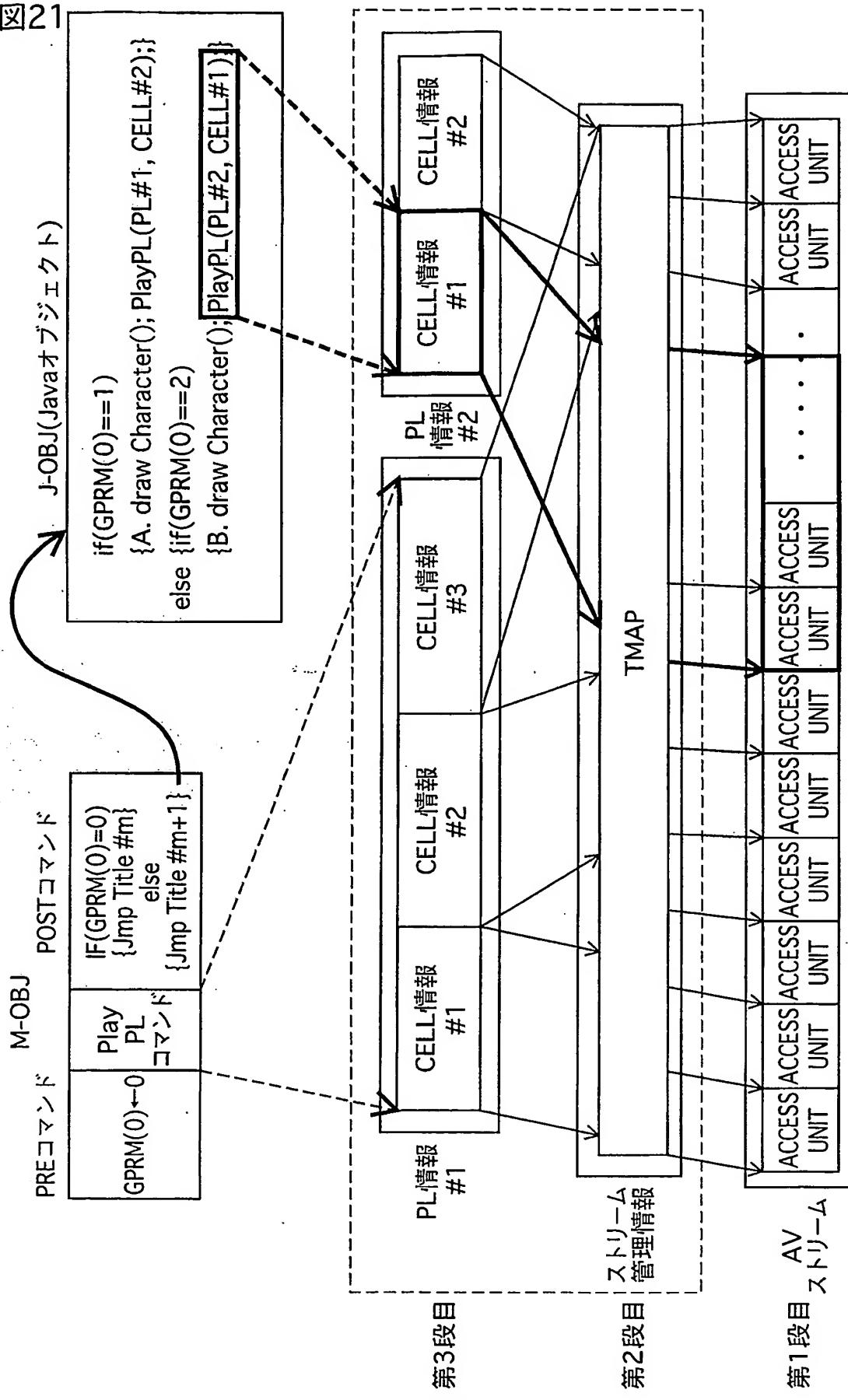


図22

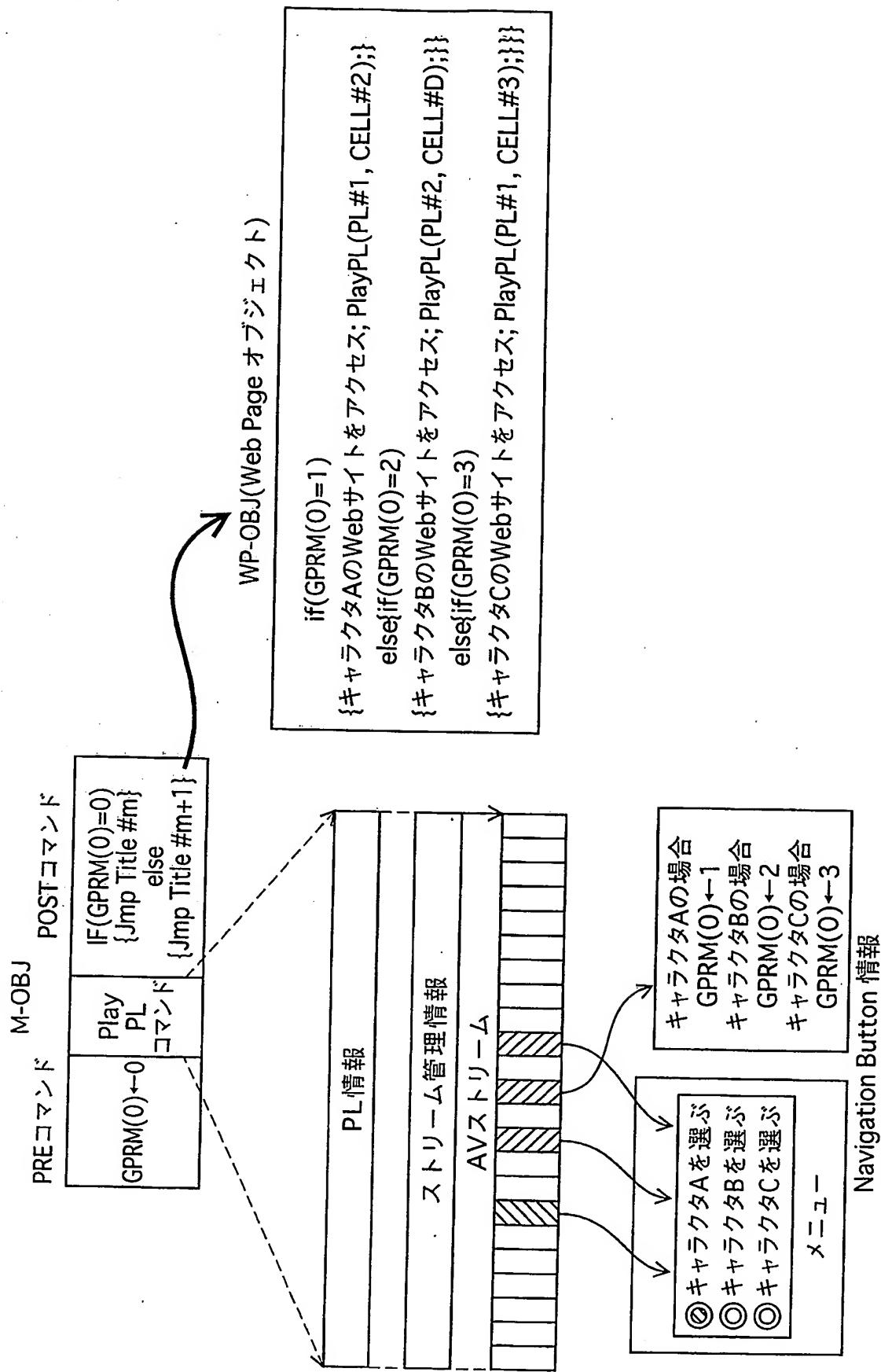


図23

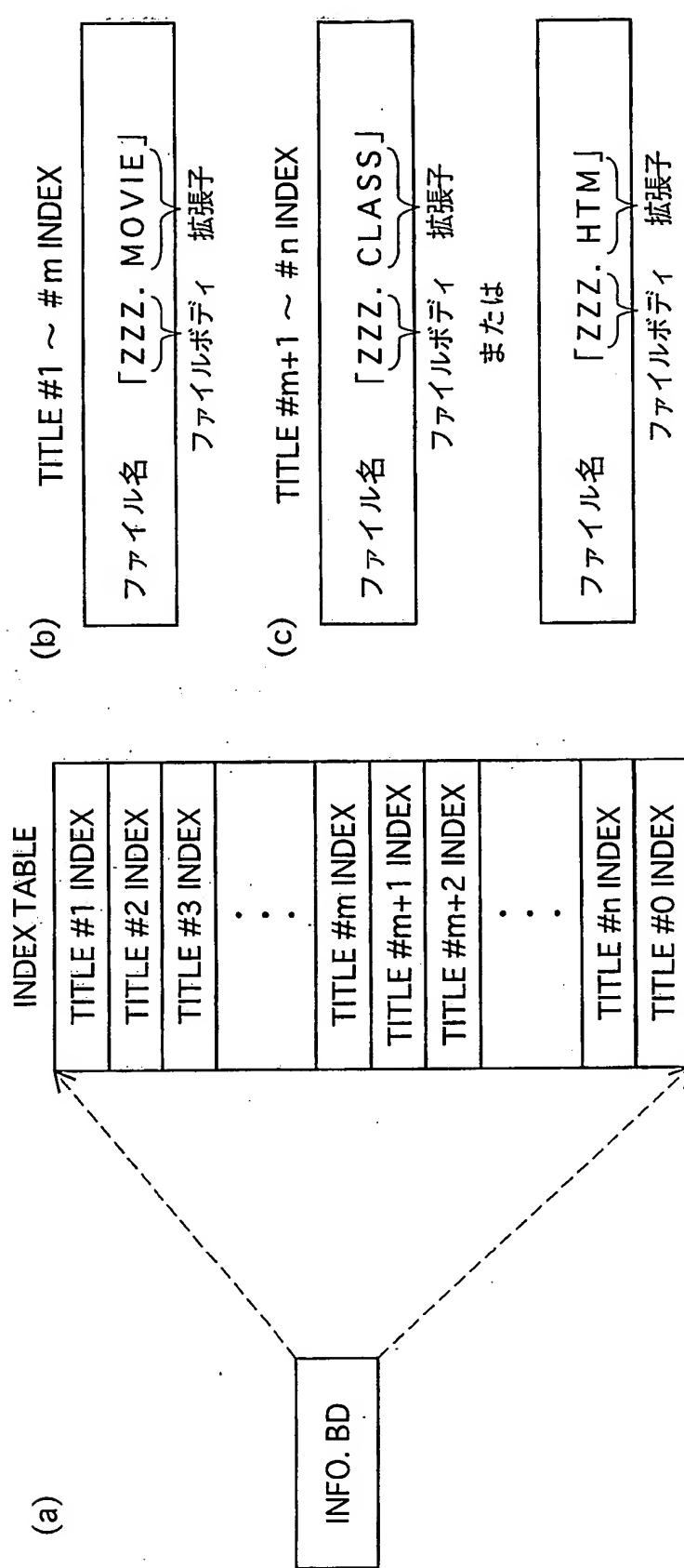
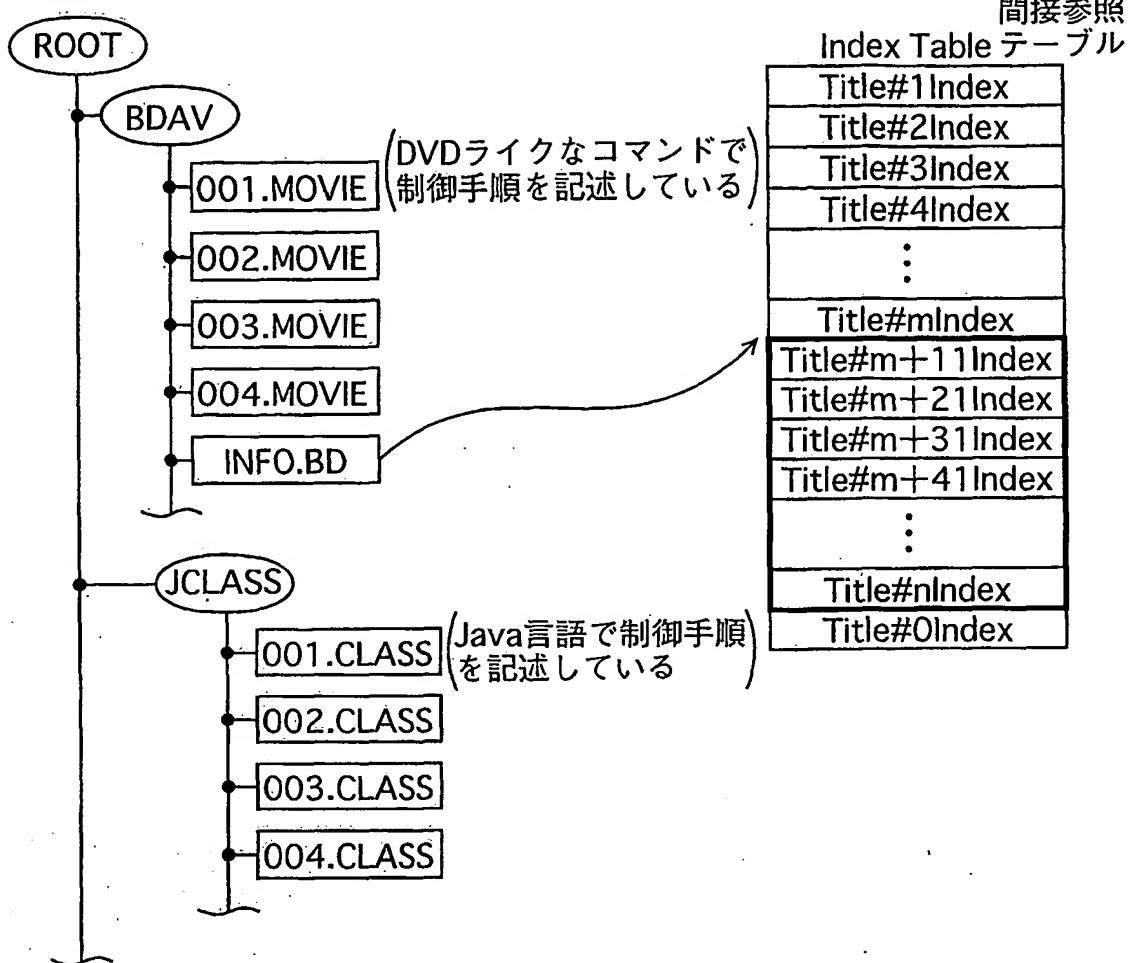


図24

(a)



(b)

Index Table	
Title#1Index	ファイル名「001.MOVIE」
Title#2Index	ファイル名「002.MOVIE」
Title#3Index	ファイル名「003.MOVIE」
Title#4Index	ファイル名「004.MOVIE」
:	:
Title#mIndex	
Title#m+1Index	ファイル名「001.CLASS」
Title#m+2Index	ファイル名「002.CLASS」
Title#m+3Index	ファイル名「003.CLASS」
Title#m+4Index	ファイル名「004.CLASS」
:	:
Title#nIndex	
Title#0Index	ファイル名「000.CLASS」

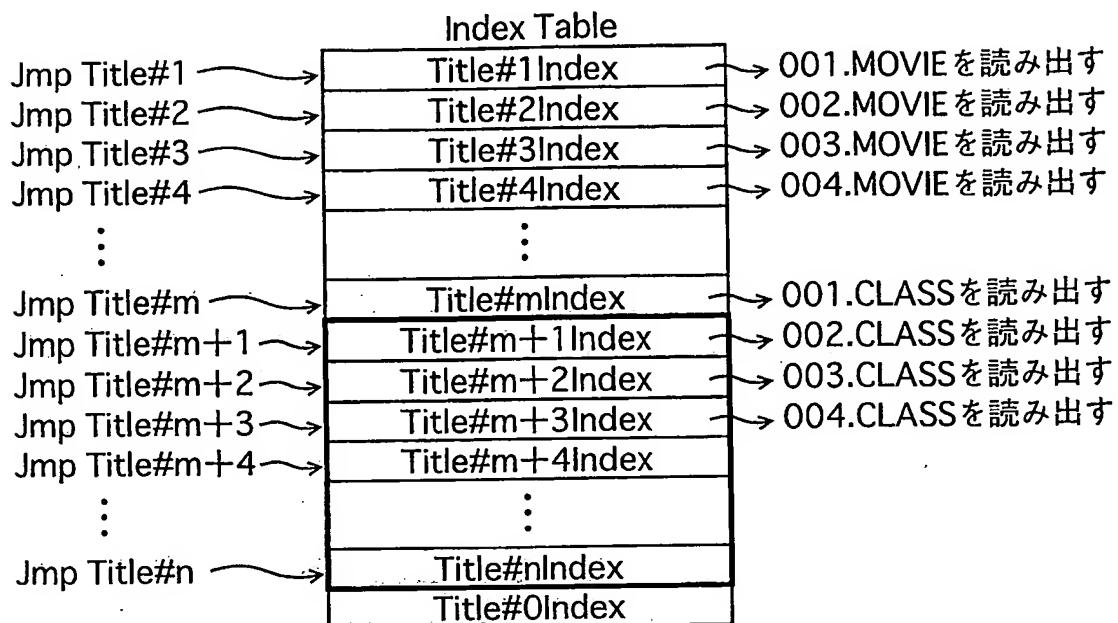
m : モードの境界値

MOVIEモード

Javaモード
(エンハンスドモード)

図25

(a) MOVIEモード+Javaモードが可能(フルシステム)



(b) MOVIEモードのみが可能(コアシステム)

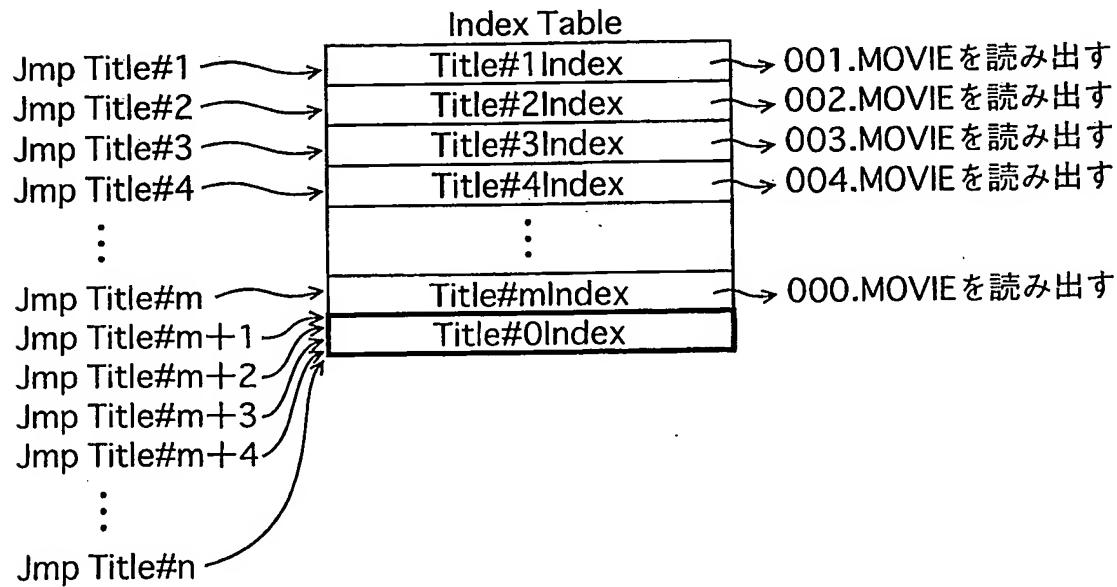


図26

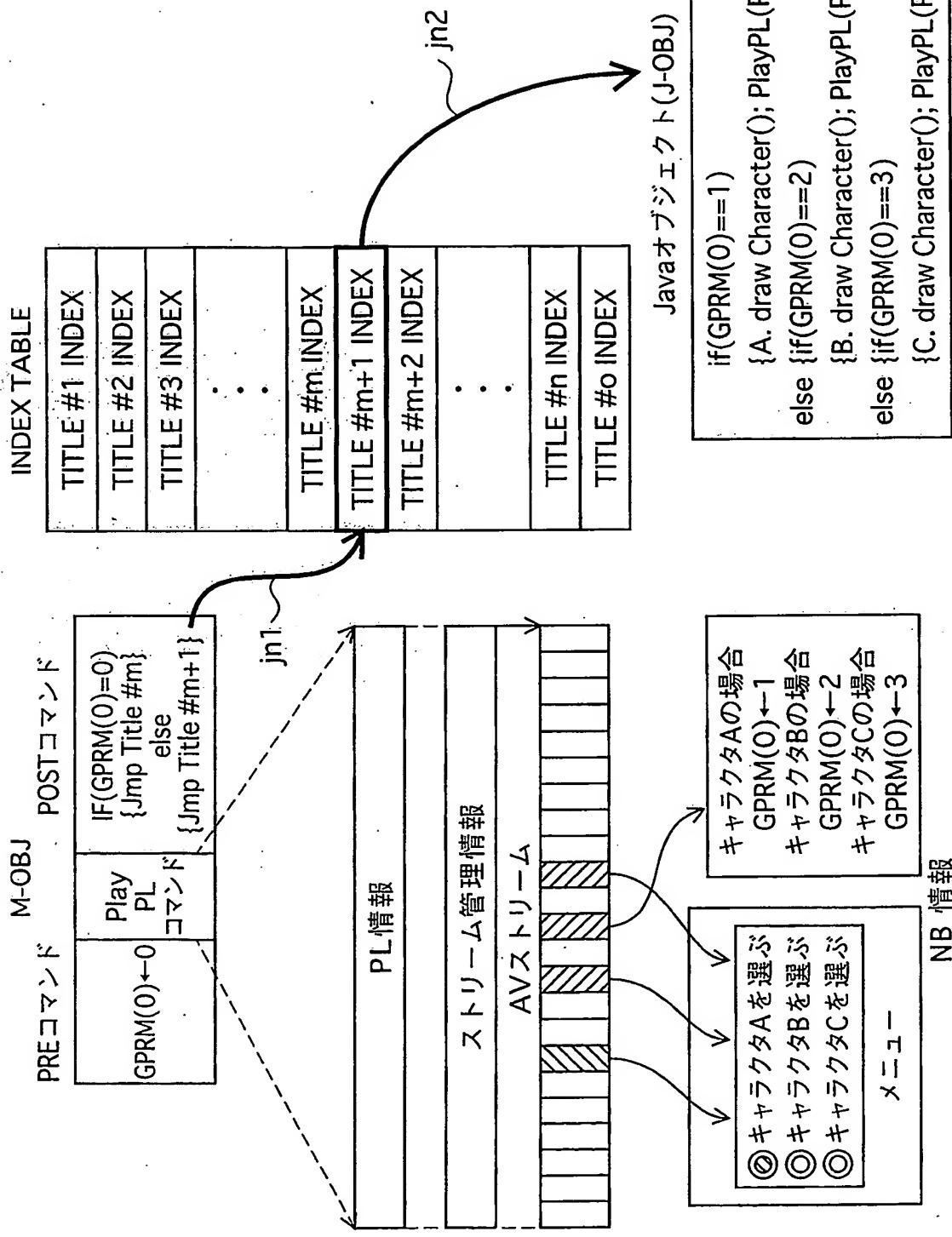


図27

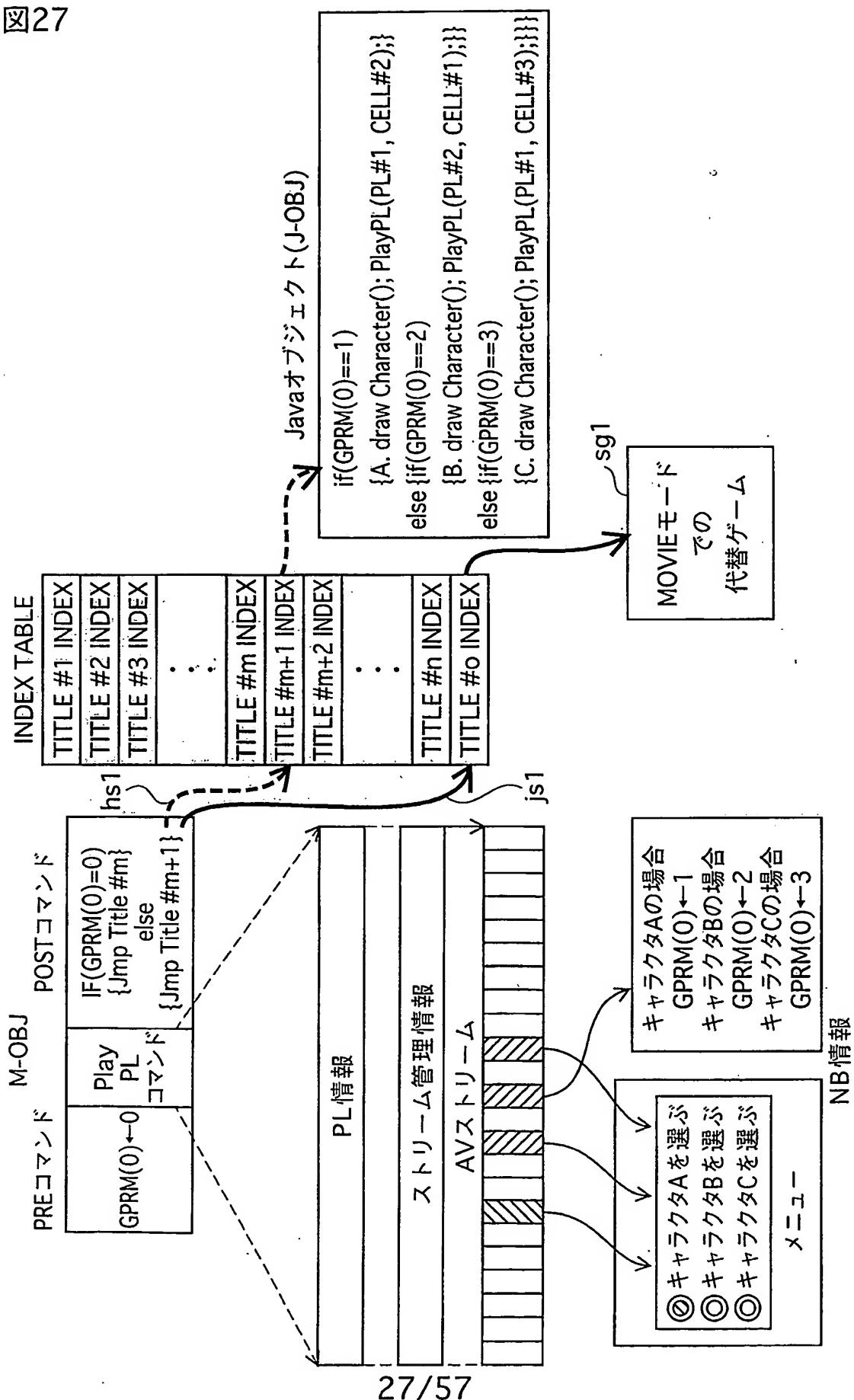


図28

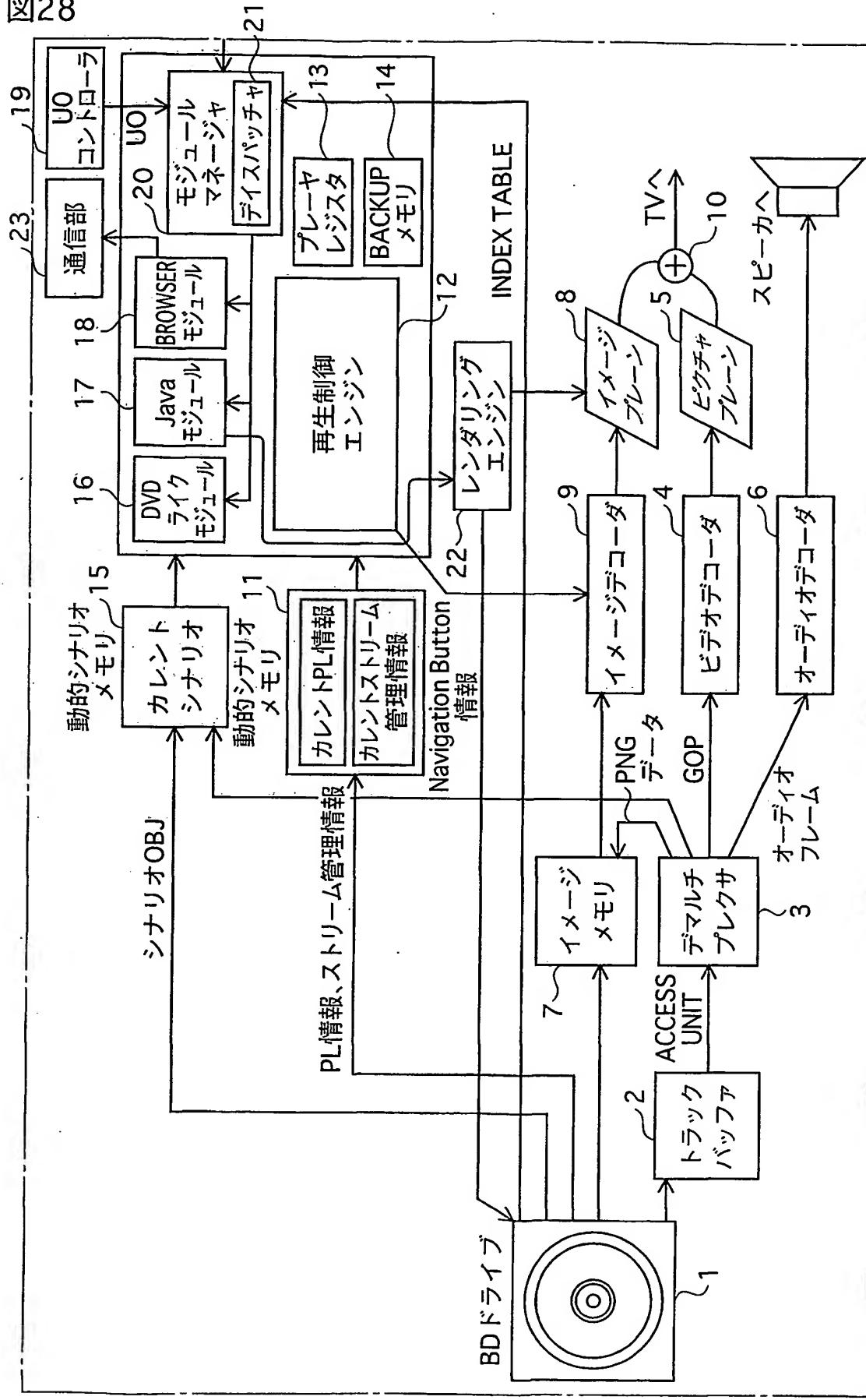


図29

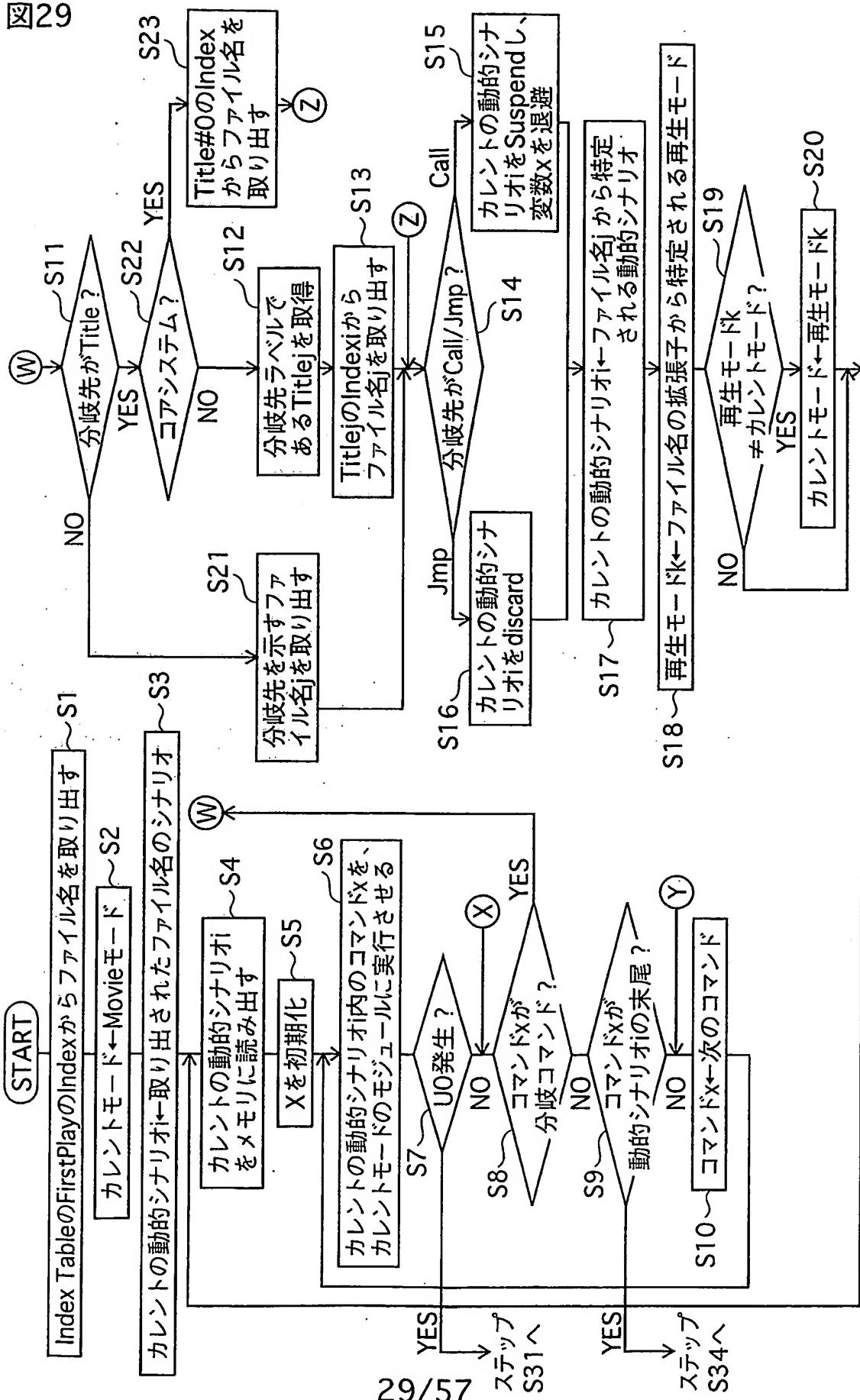


図30

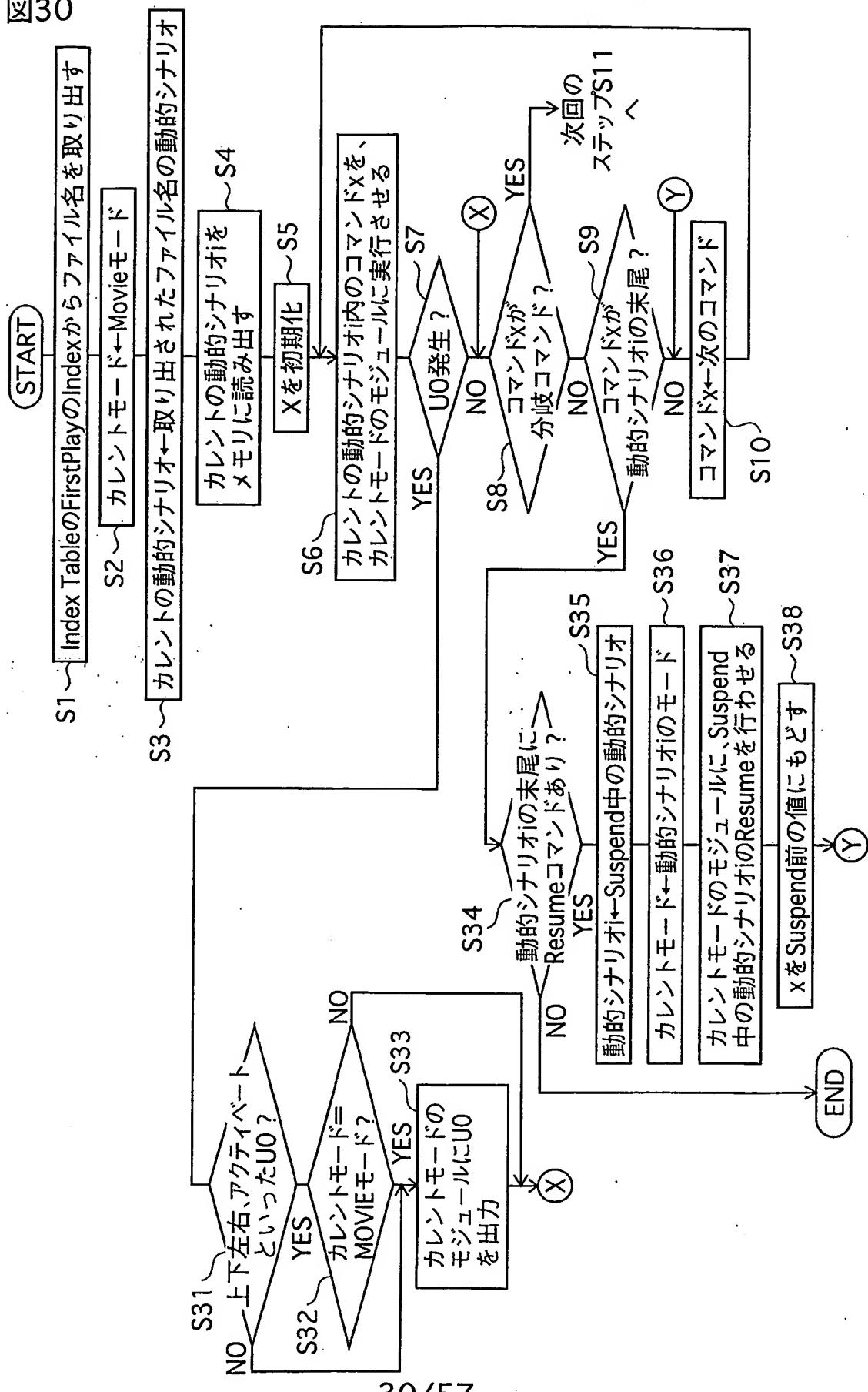


図31

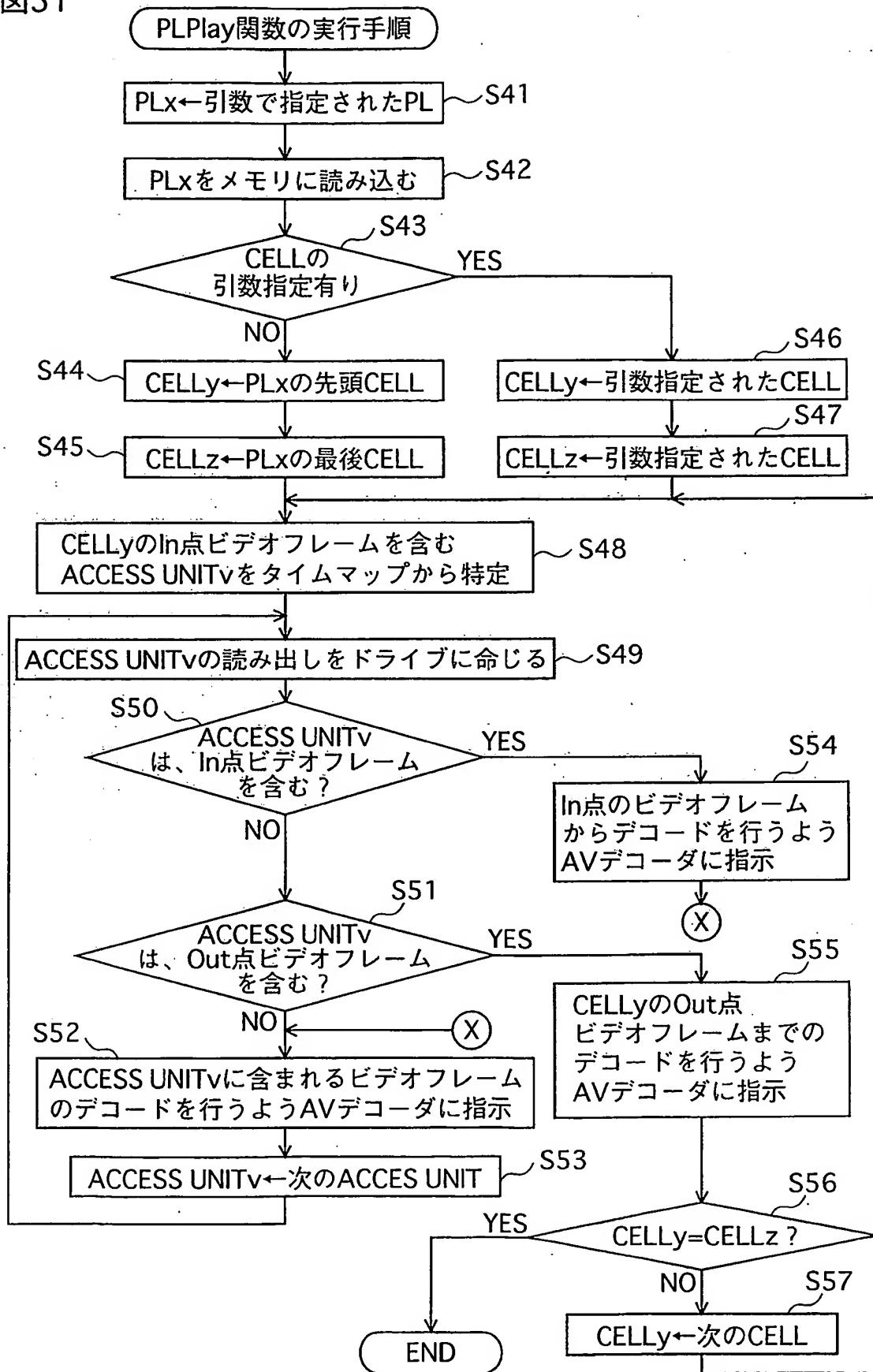


図32

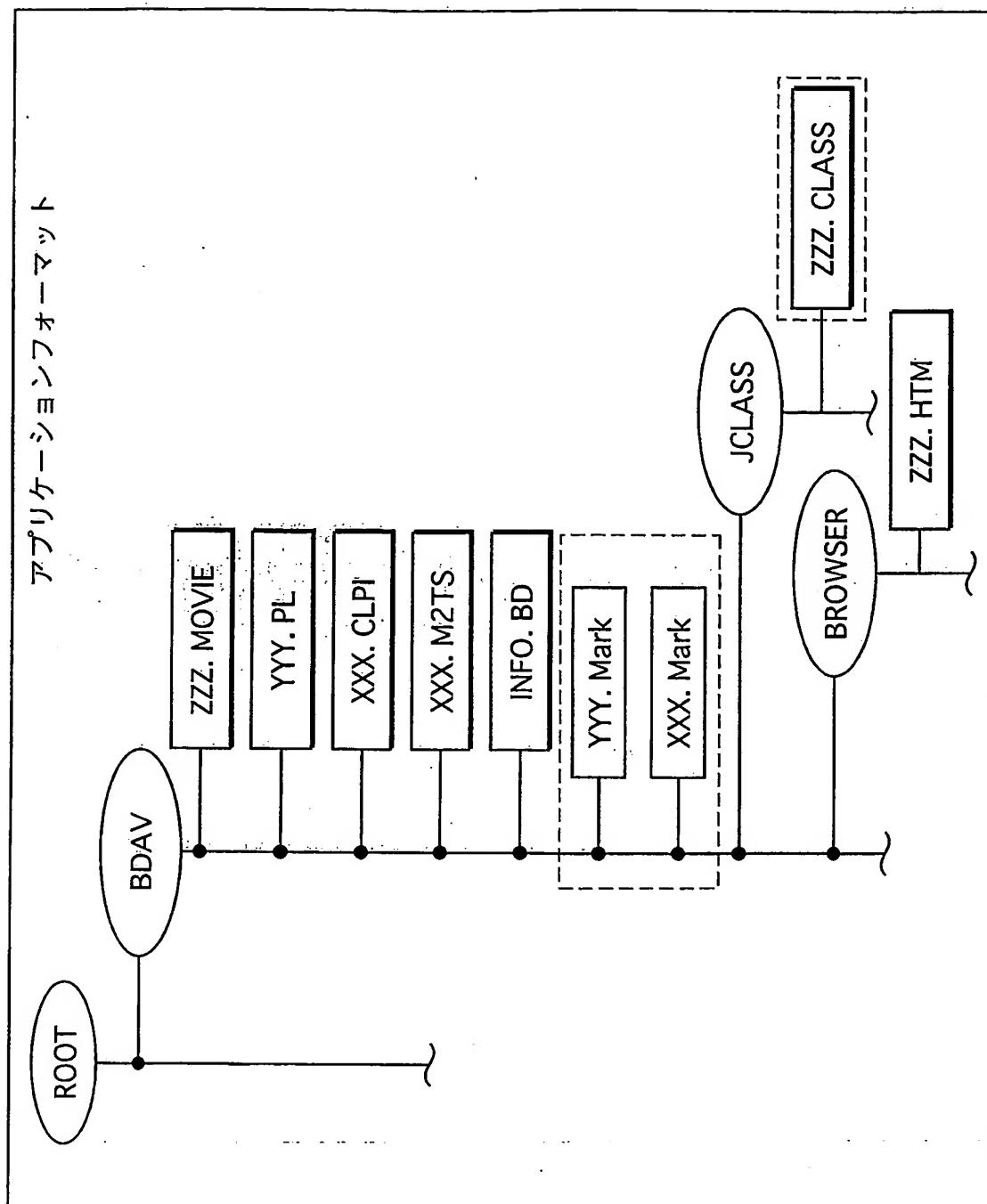


図33

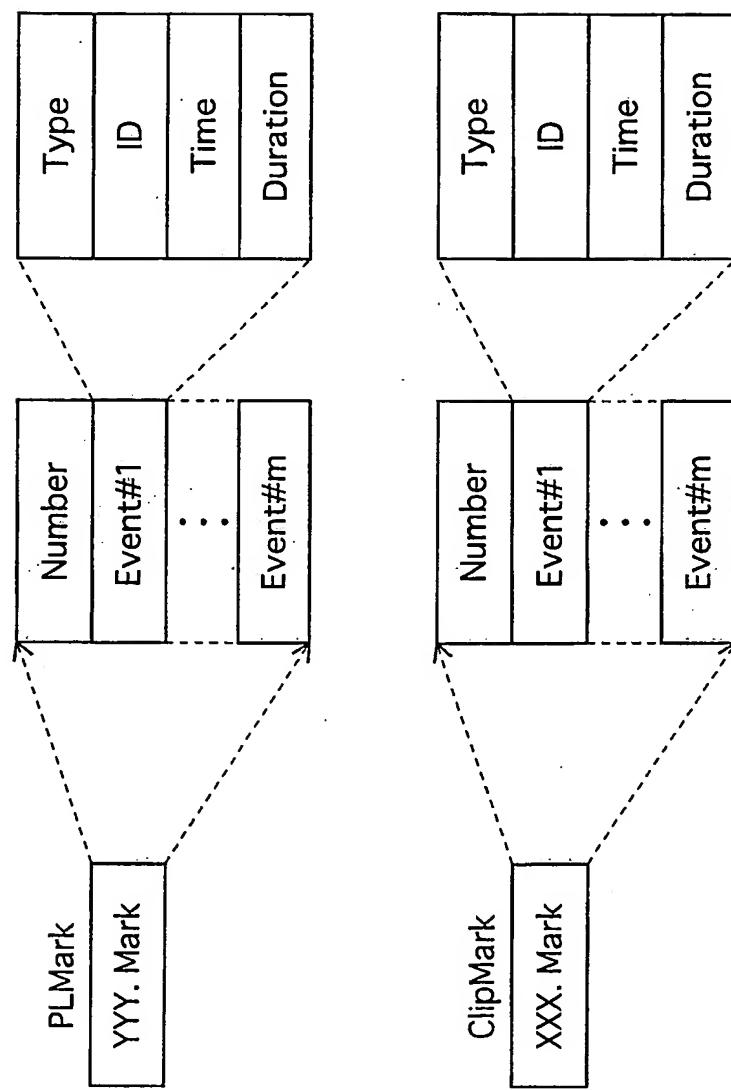
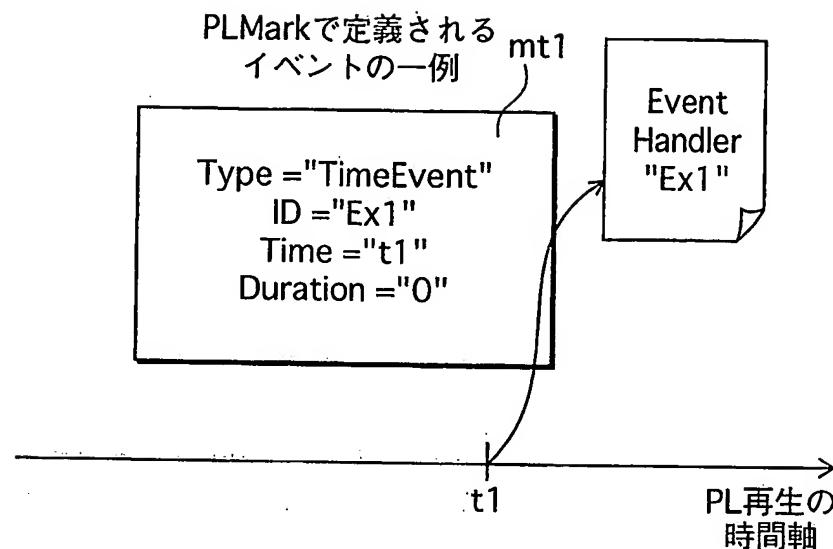


図34

(a)



(b)

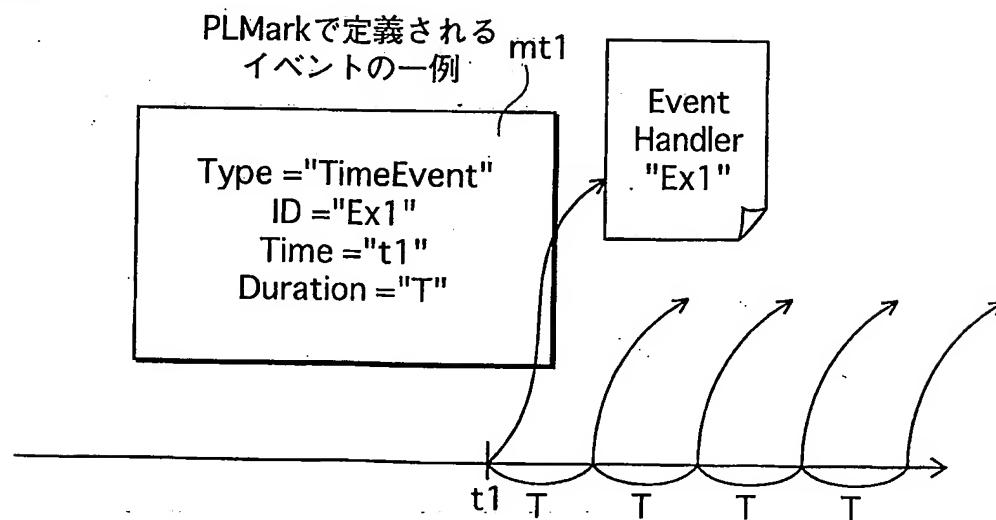


図35

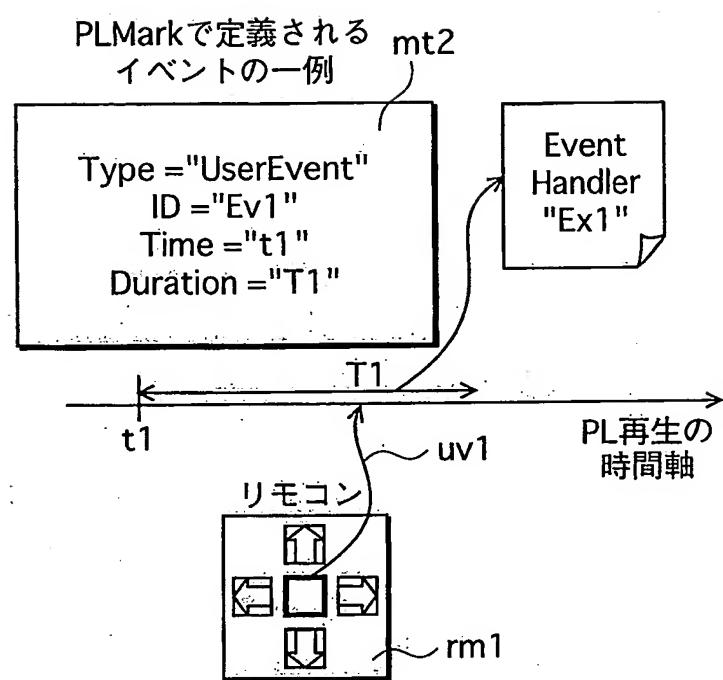


図36

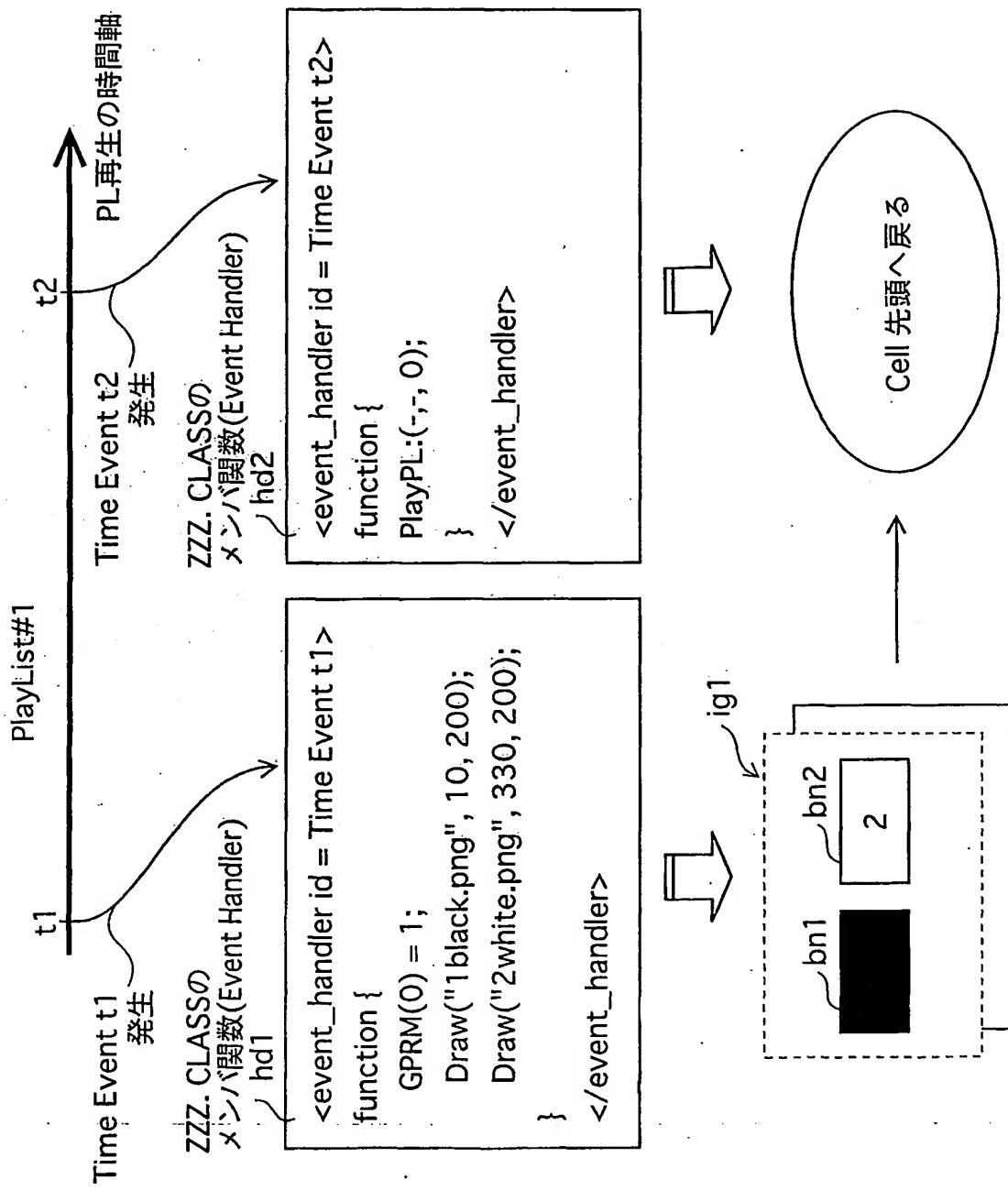


図37

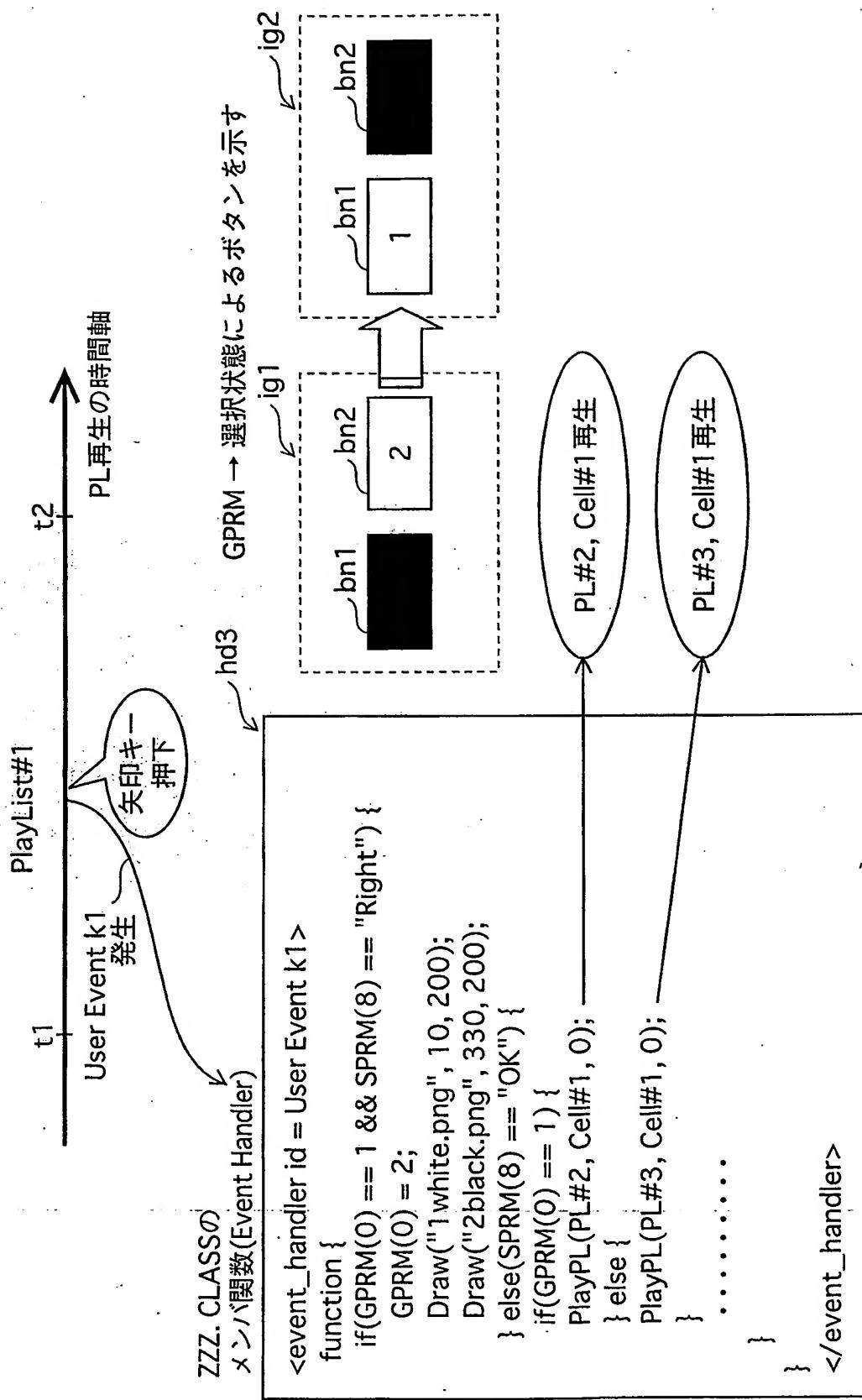


図38

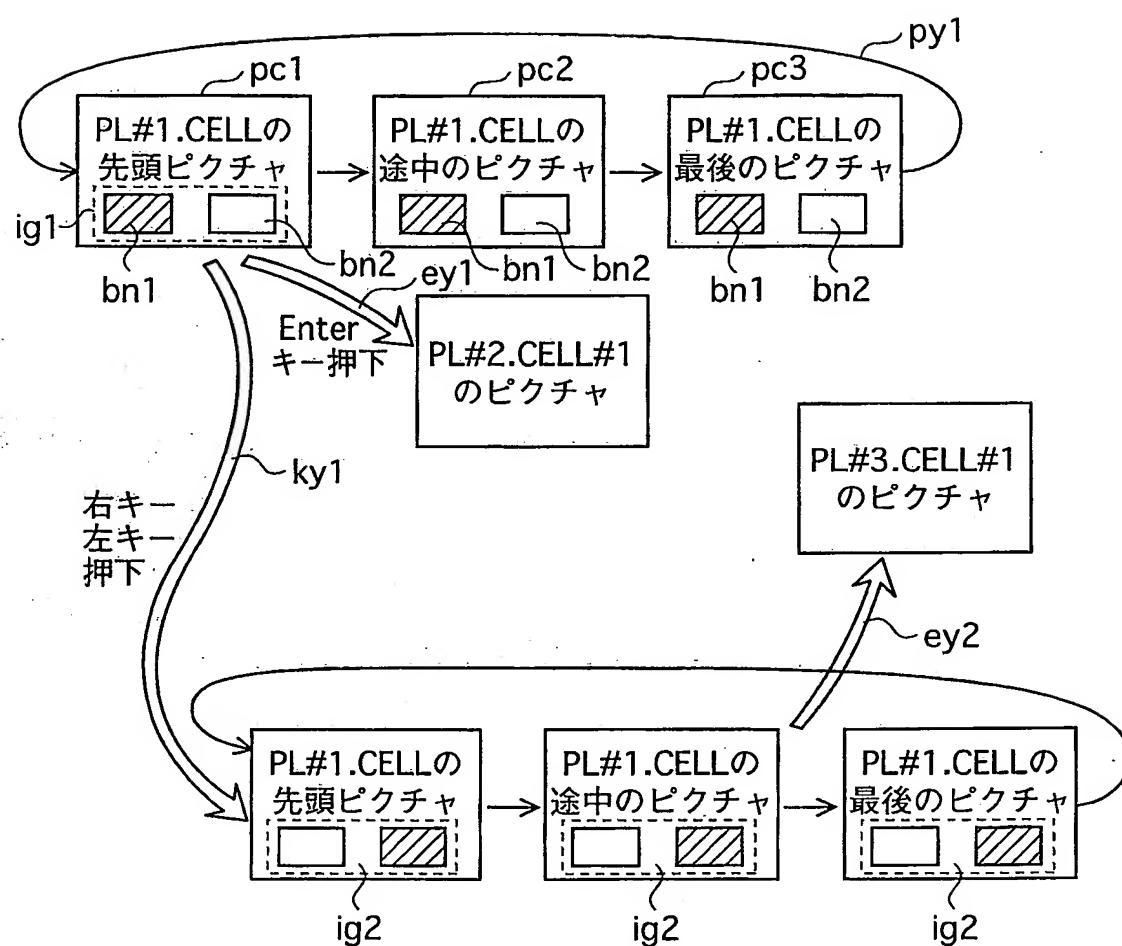


図39

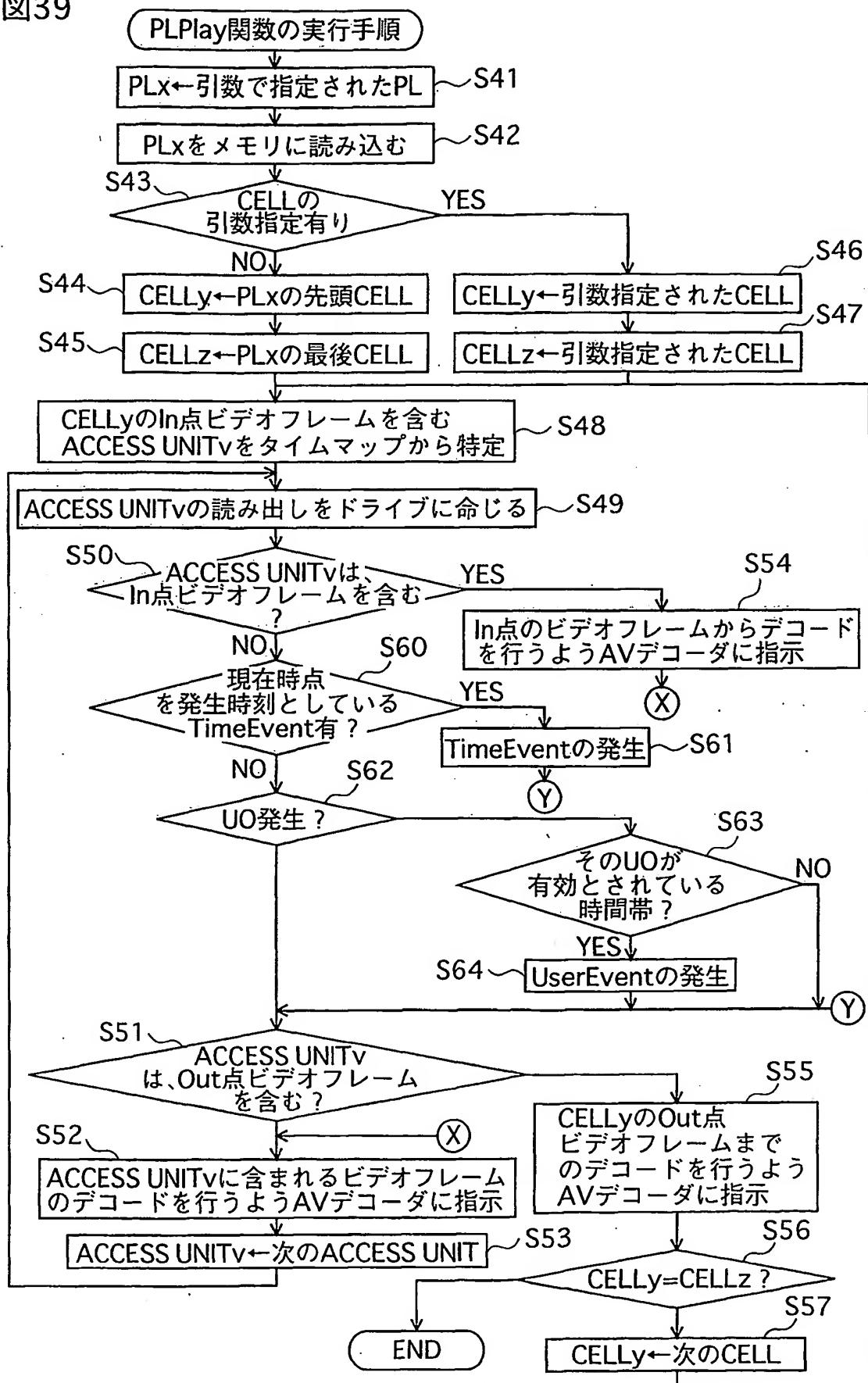


図40

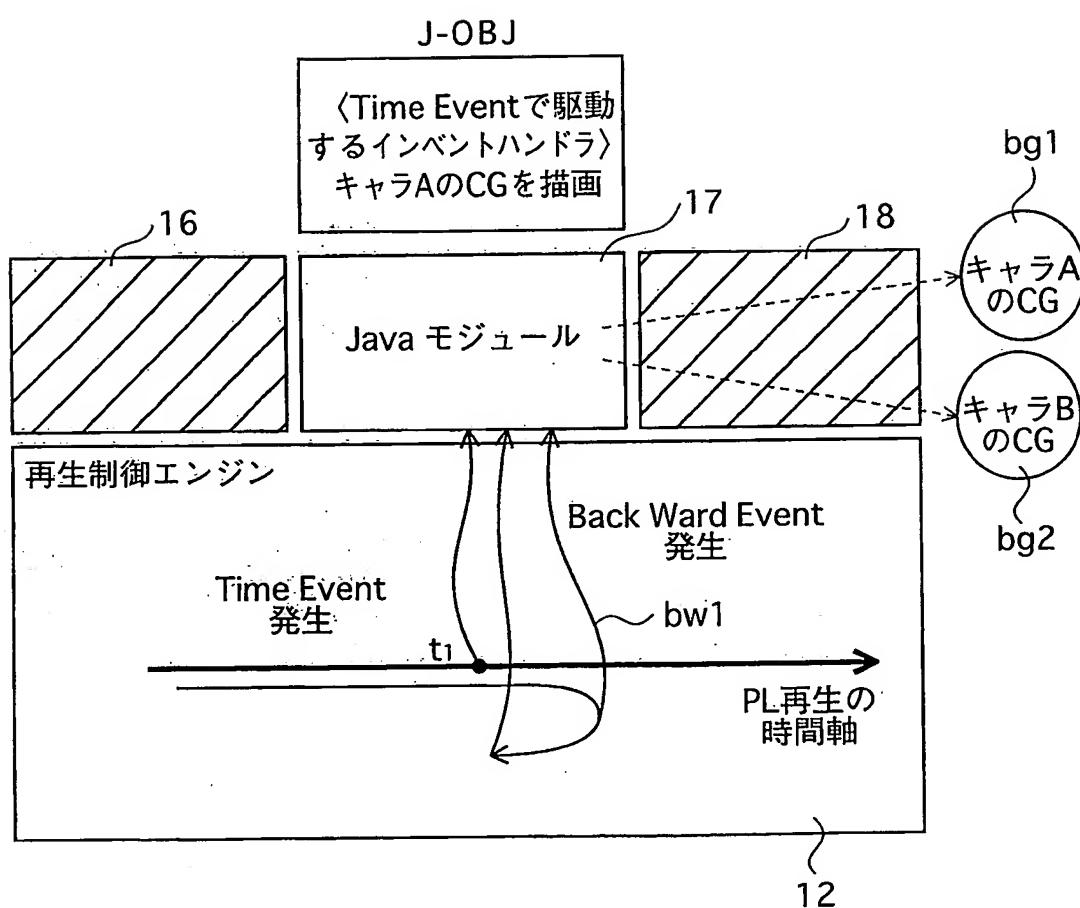


図41

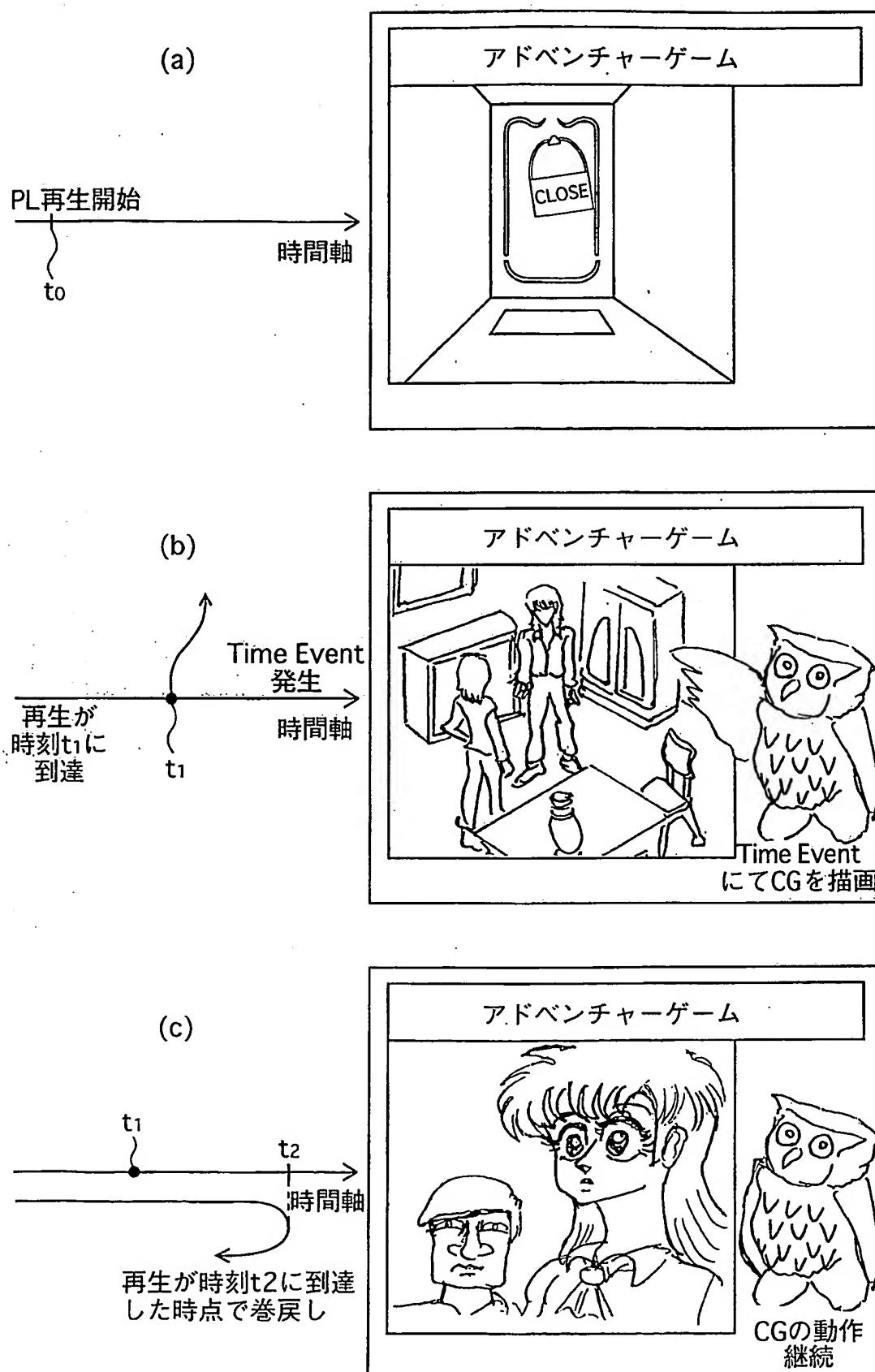
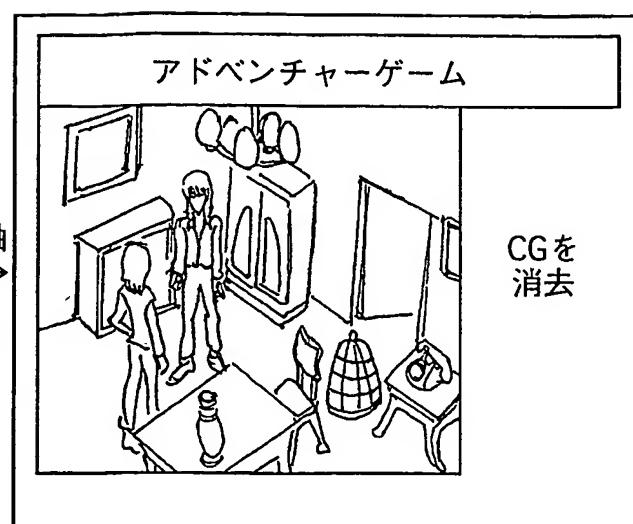
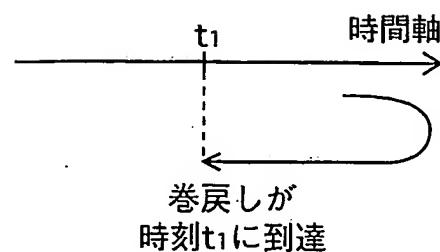
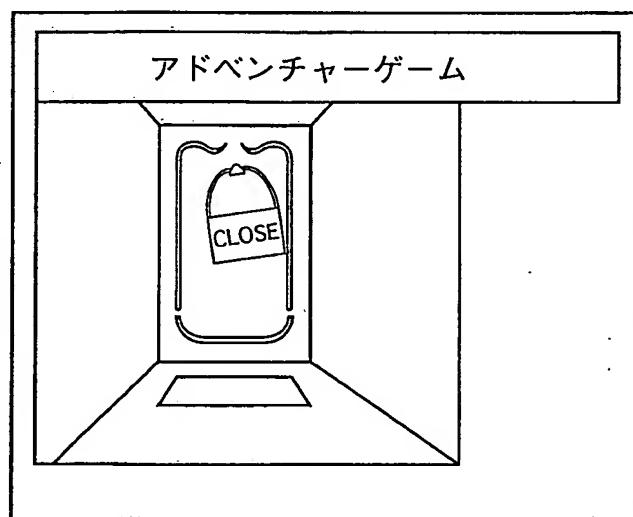
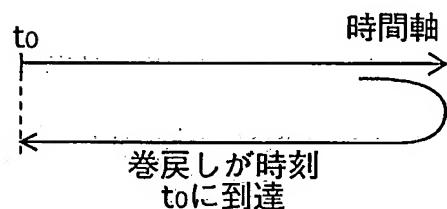


図42

(a)



(b)



(c)

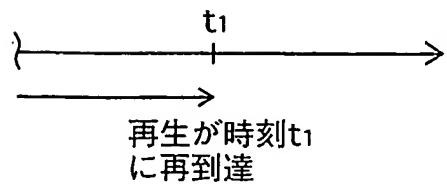


図43

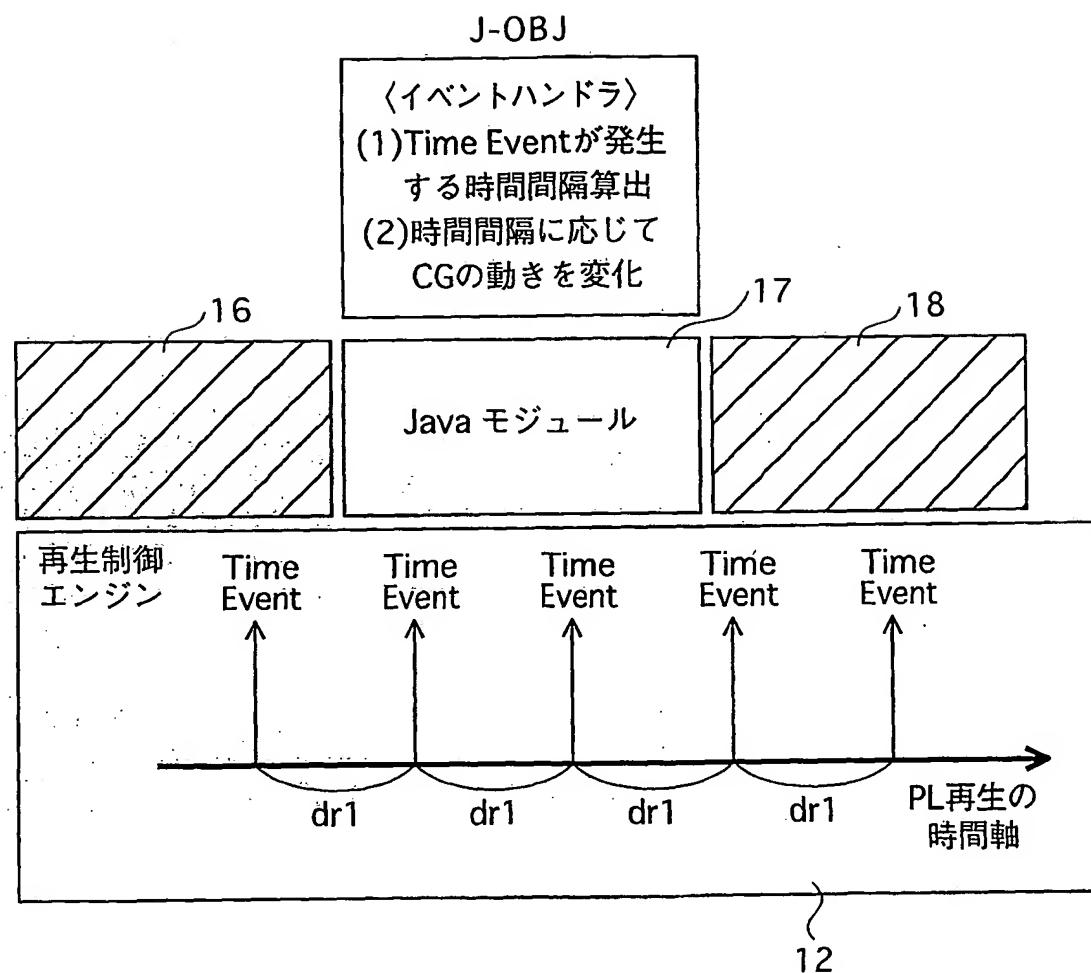


図44

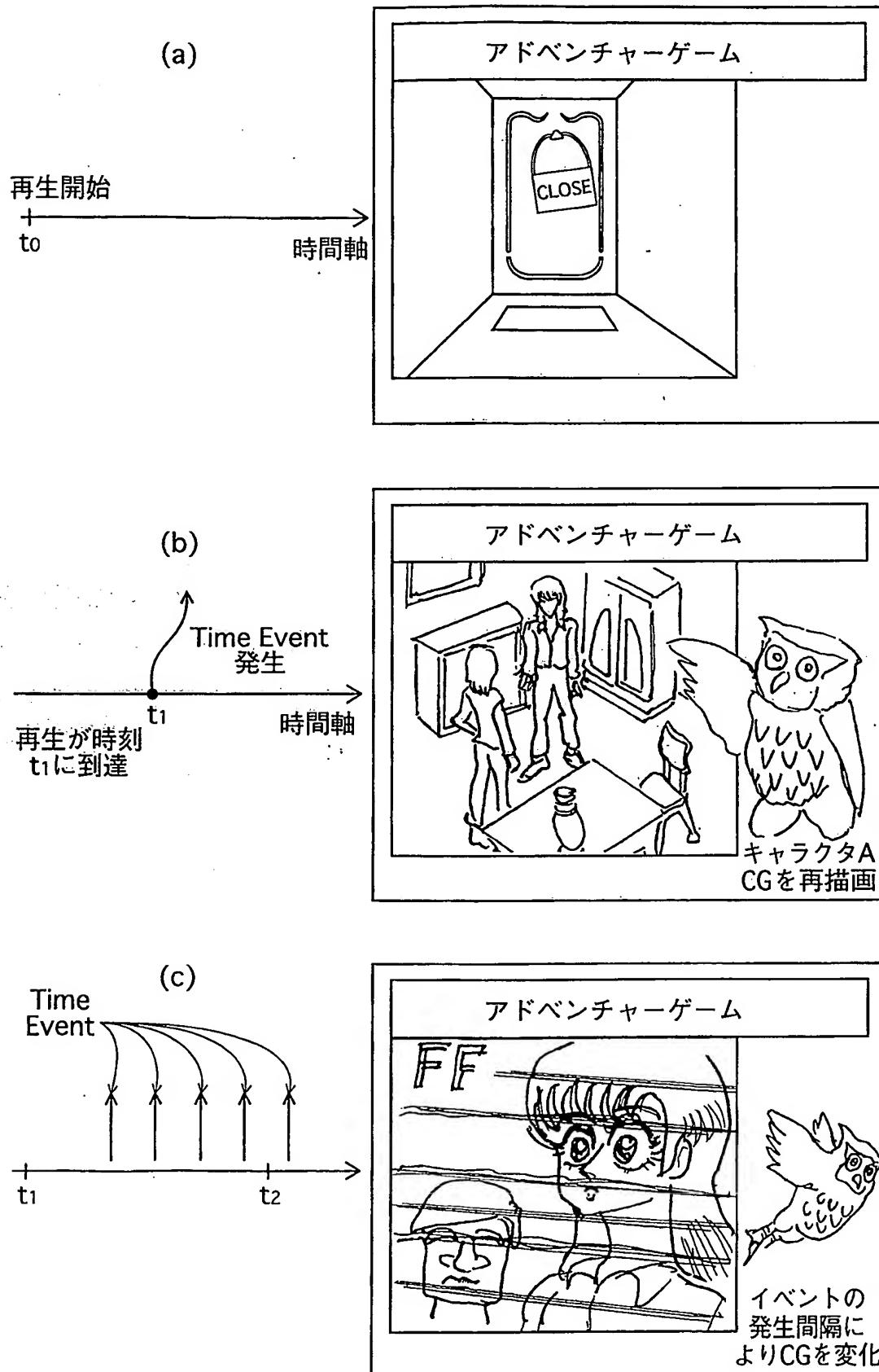


図45

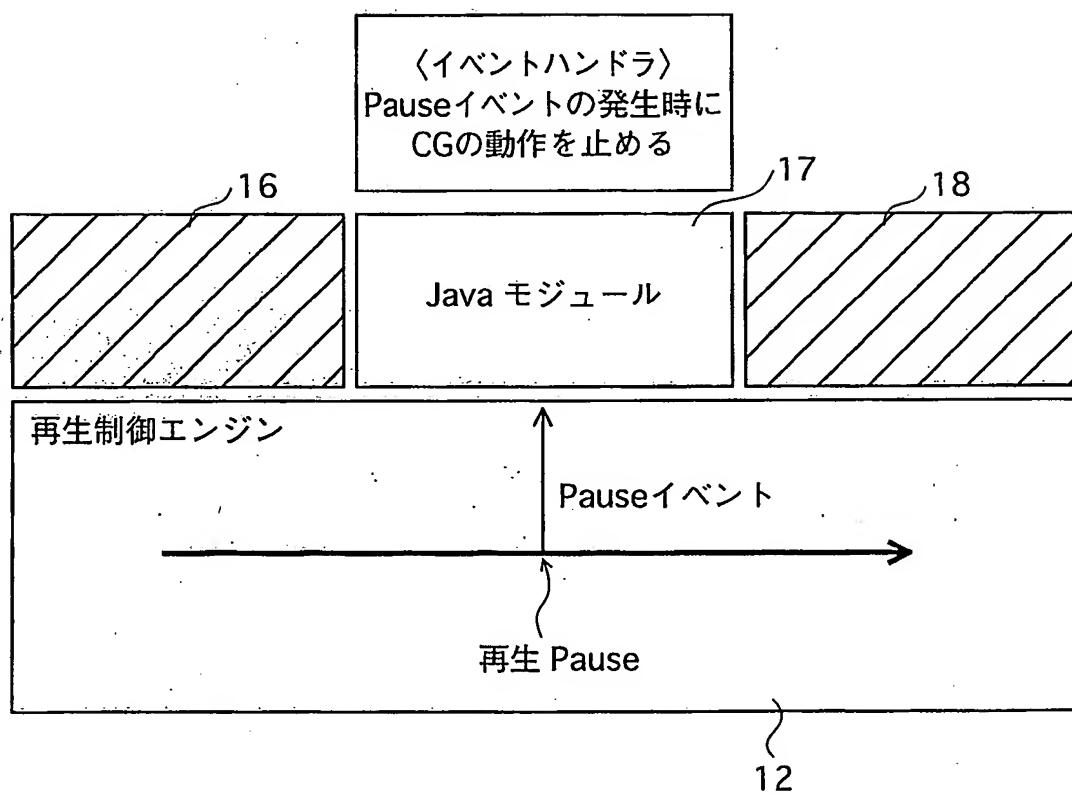


図46

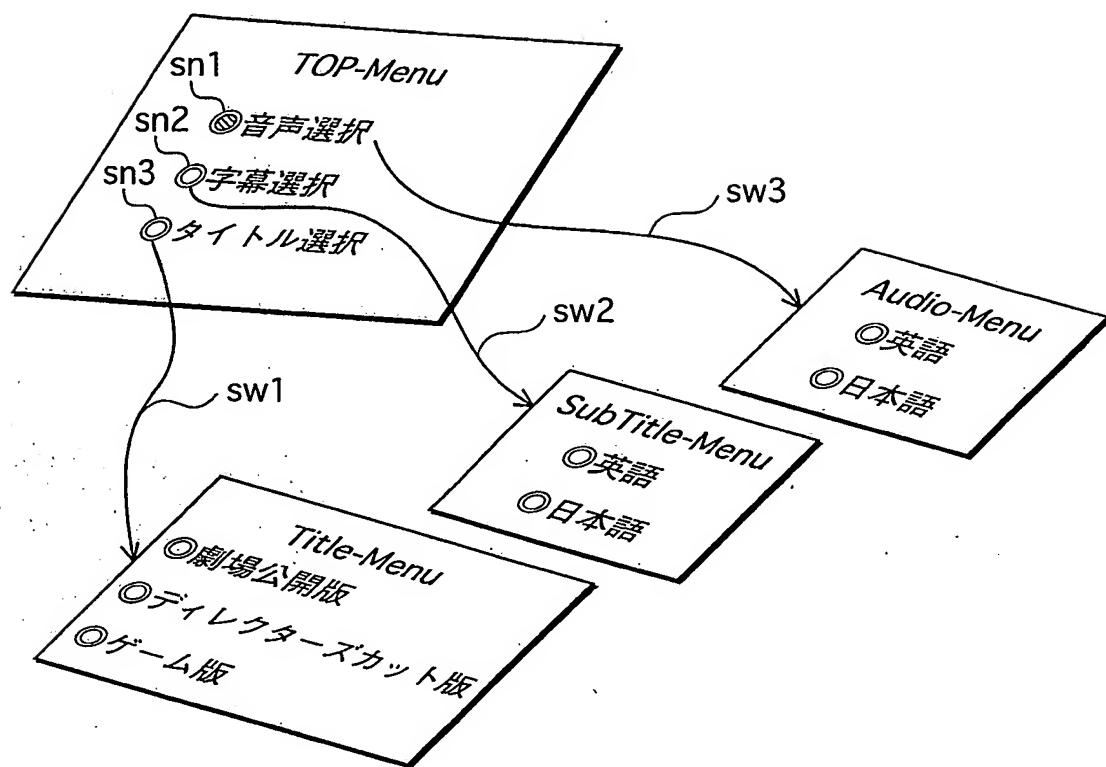


図47

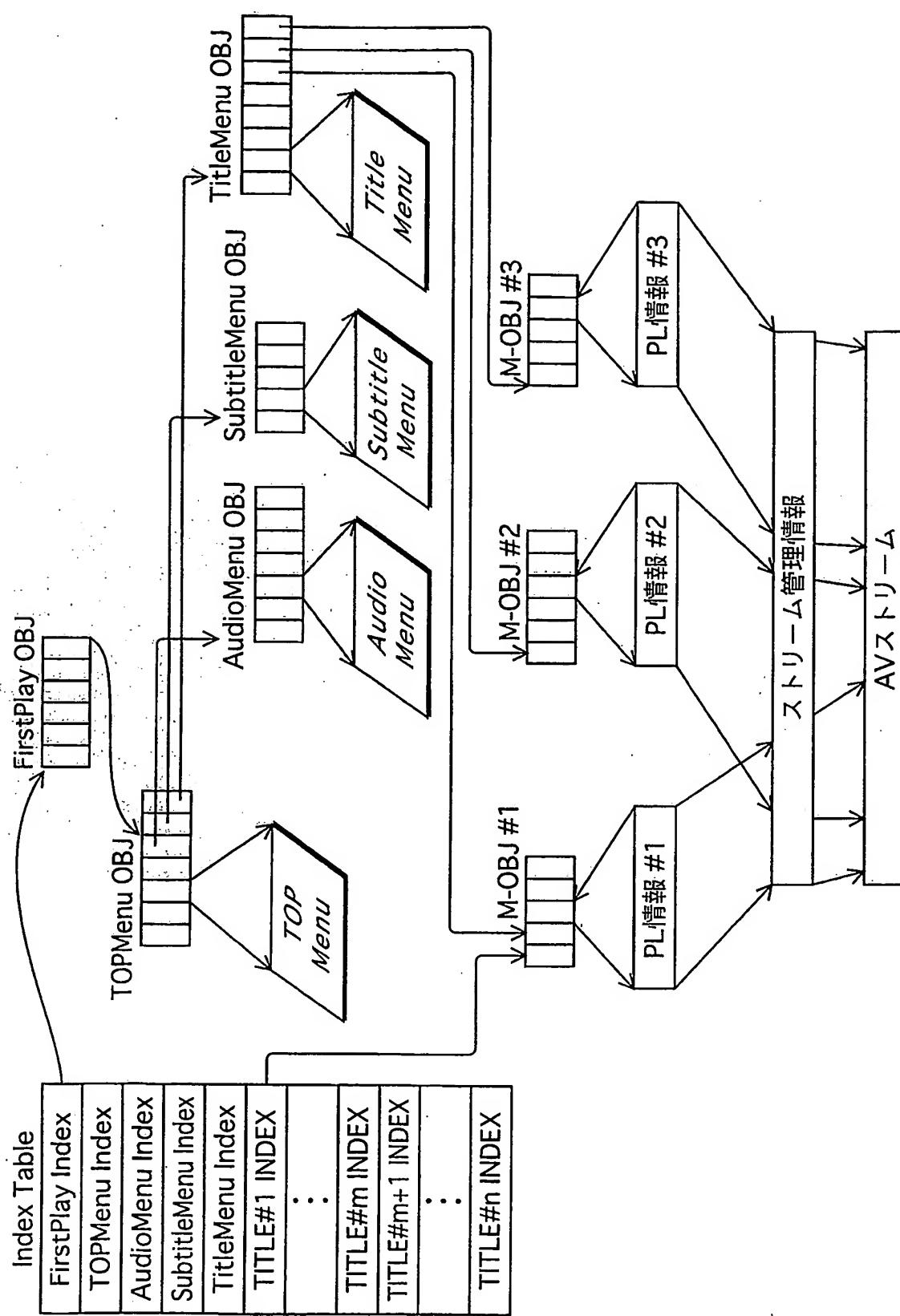


図48

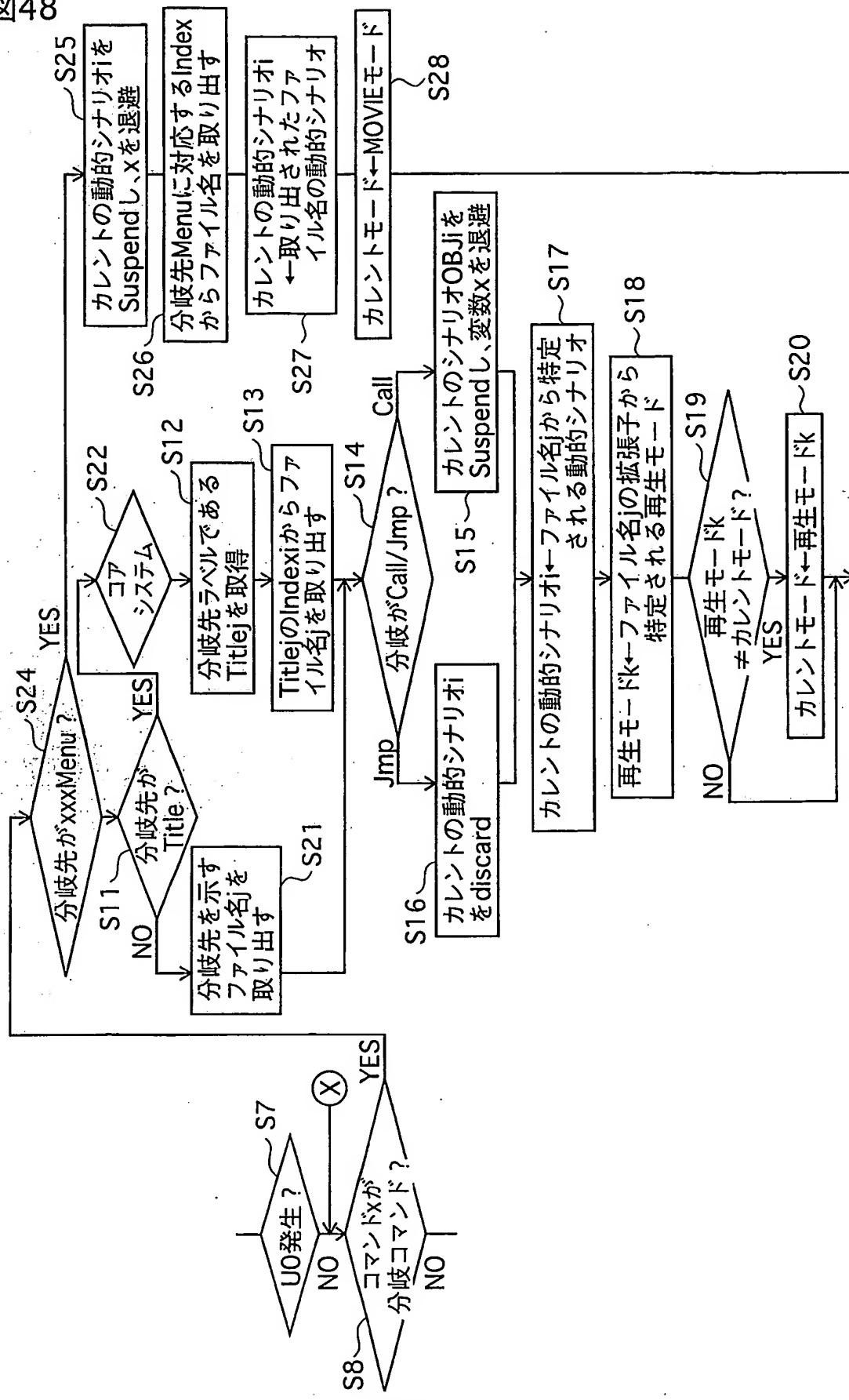


図49

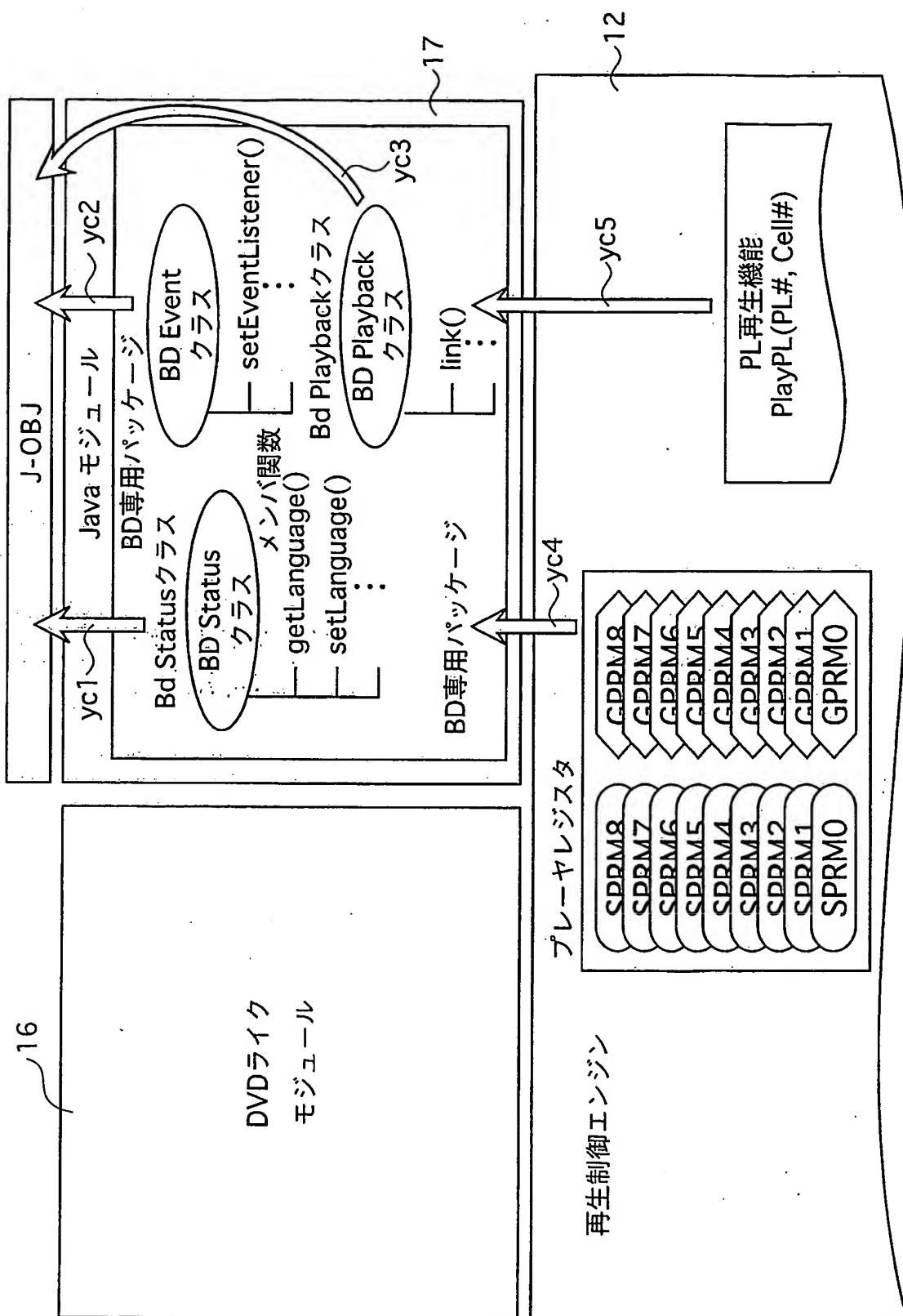


図50

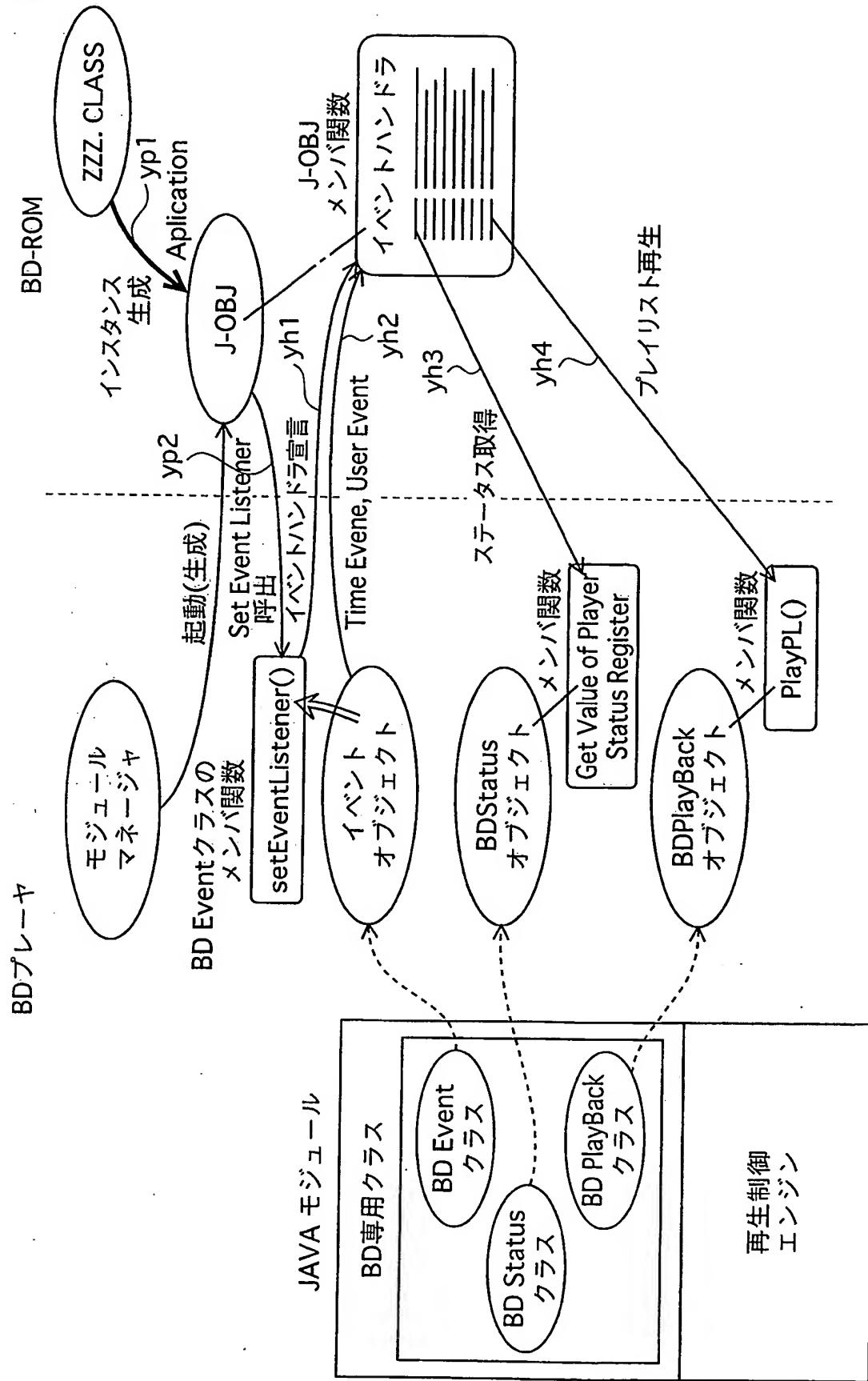


図51

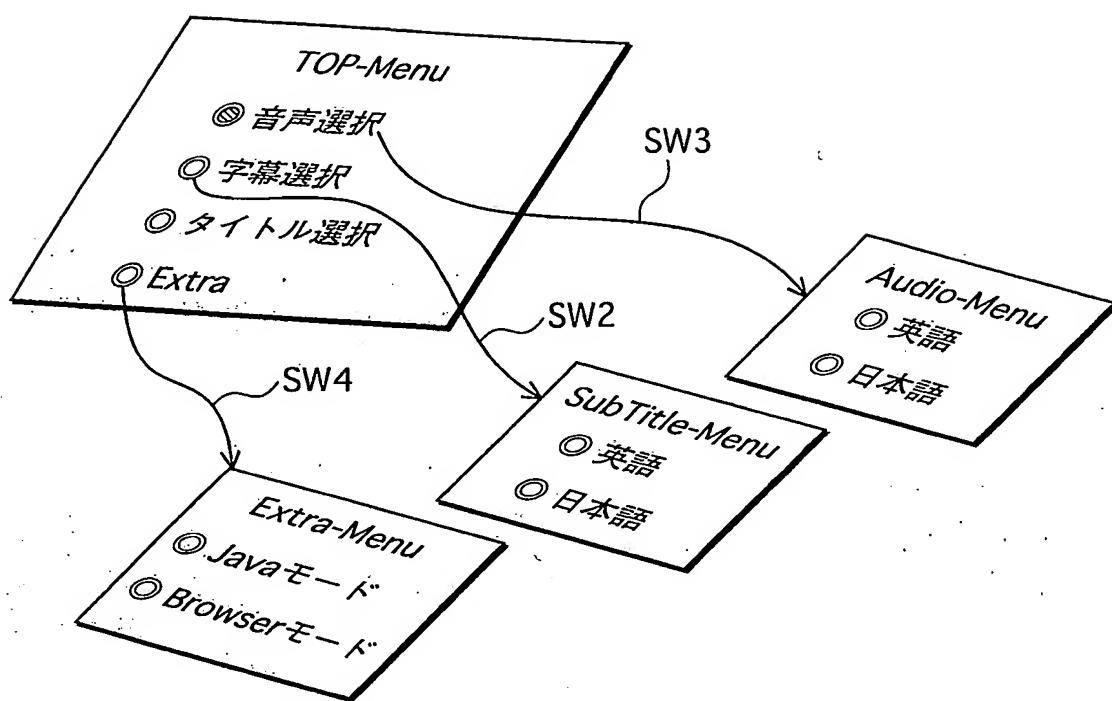


図52

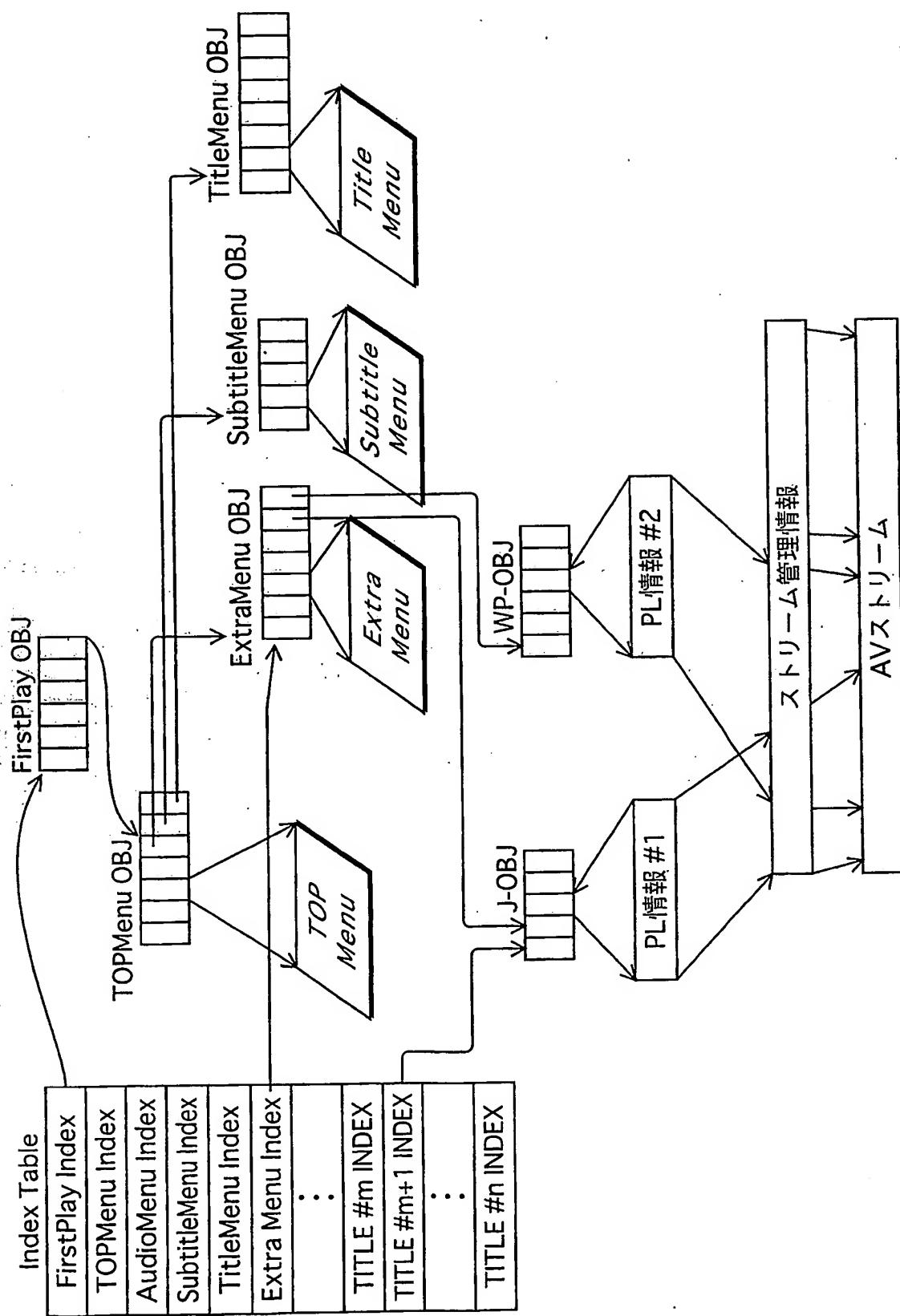


図53

J-OBJが用いる
スケジュール テーブル

キャラクタA	03:00~05:00, PL#1, CELL#1
キャラクタB	05:01~08:00, PL#1, CELL#1
キャラクタC	08:01~10:00, PL#1, CELL#2
キャラクタAとB	10:01~17:00, PL#1, CELL#2

図54

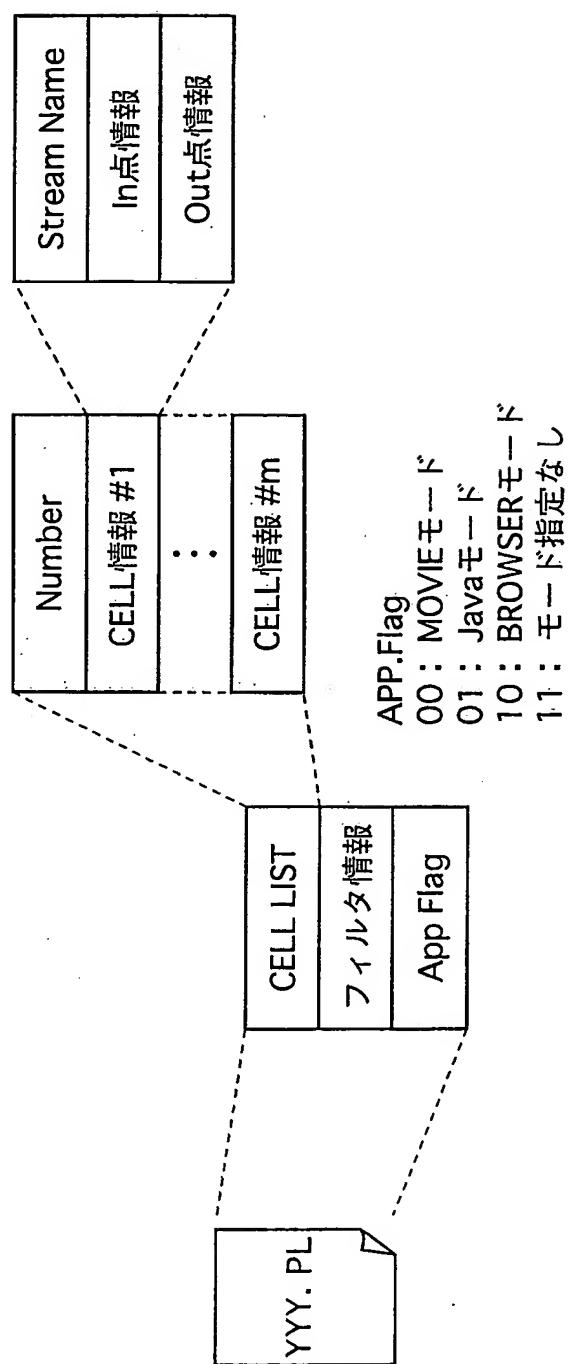


図55

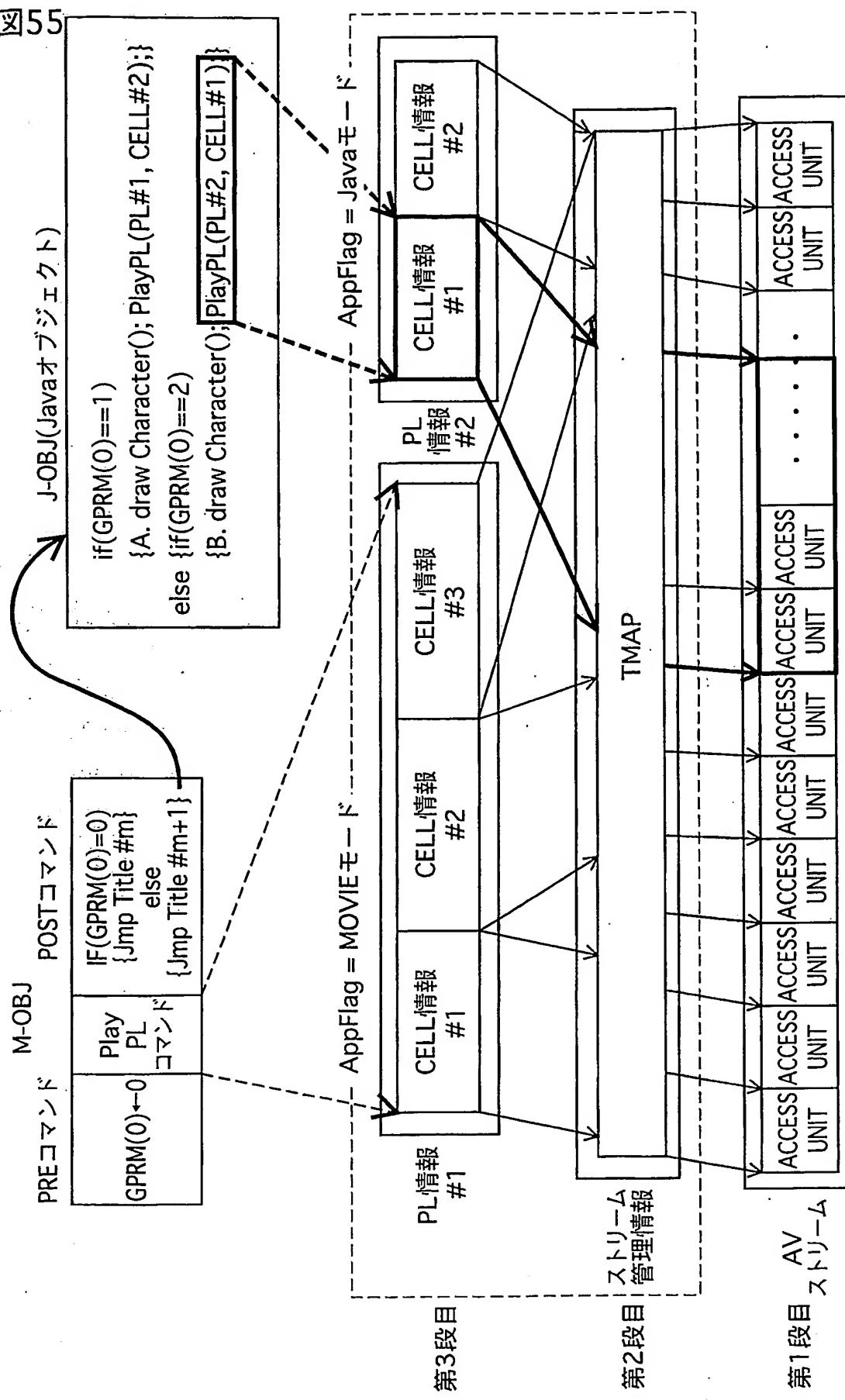


図56

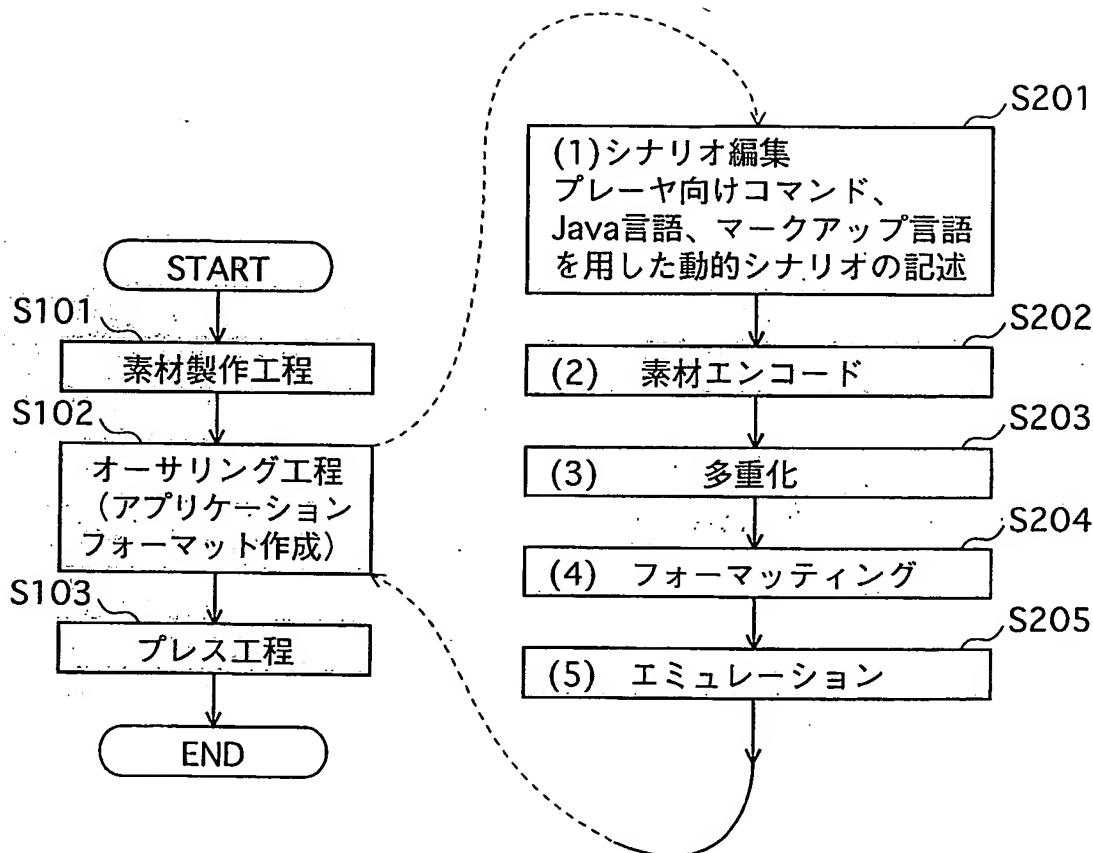
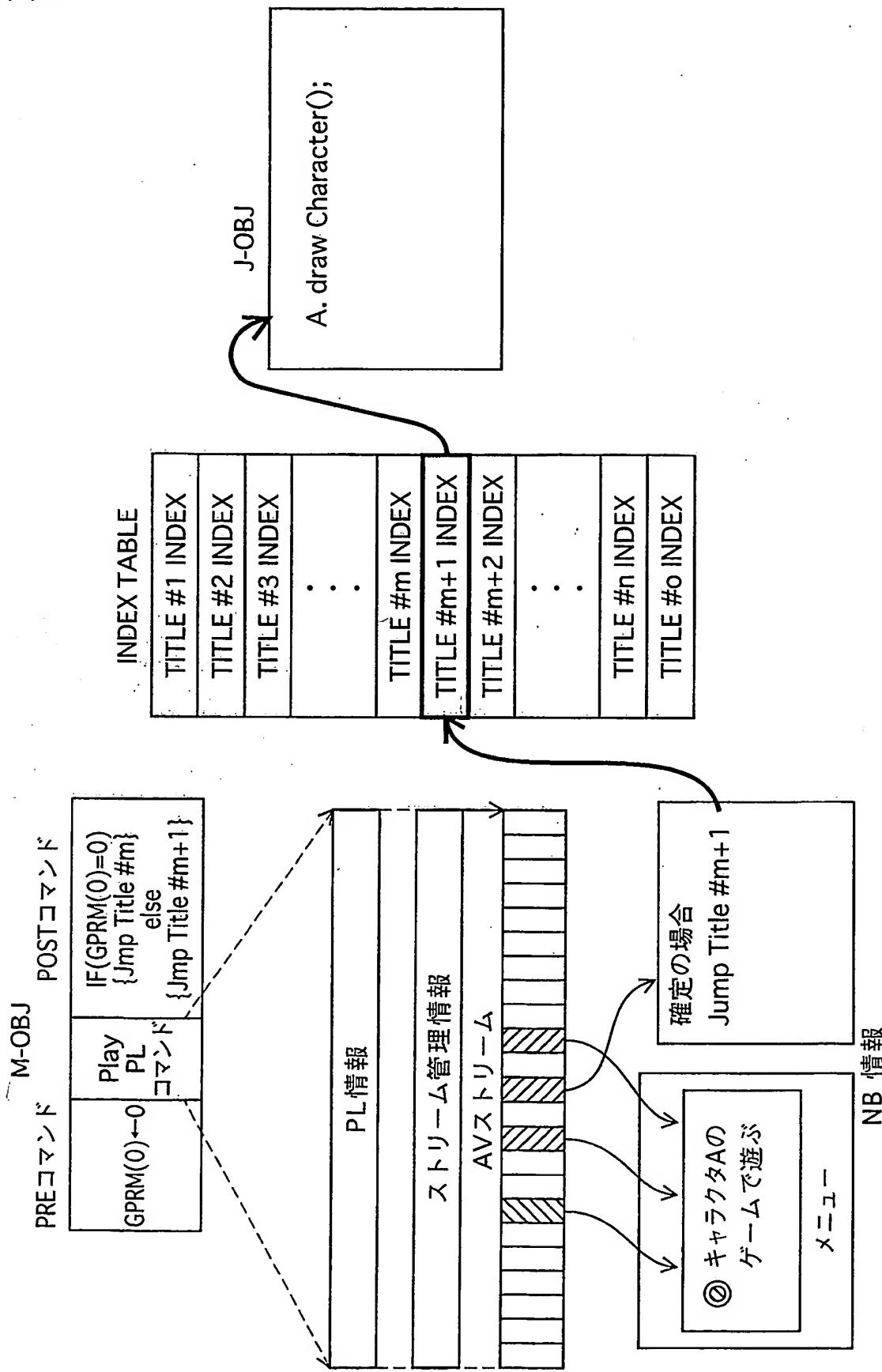


図57



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP03/11679

A. CLASSIFICATION OF SUBJECT MATTER

Int.Cl⁷ G11B27/00, G11B27/10, G11B20/10, G11B20/12

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

Int.Cl⁷ G11B27/00-27/06, G11B27/10, G11B20/10, G11B20/12

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Jitsuyo Shinan Koho	1922-1996	Jitsuyo Shinan Toroku Koho	1996-2003
Kokai Jitsuyo Shinan Koho	1971-2003	Toroku Jitsuyo Shinan Koho	1994-2003

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	JP 2000-57746 A (Toshiba Corp., Toshiba A.V.E. Kabushiki Kaisha), 25 February, 2000 (25.02.00), Par. Nos. [0030] to [0088], [0210] to [0246]; Figs. 12 to 18 & US 2001/0040862 A1	1-2, 23-25, 45-47 4-22, 27-44 3, 26
Y	JP 11-161663 A (Toshiba Corp.), 18 June, 1999 (18.06.99), Par. Nos. [0018] to [0020], [0030] to [0031], [0055] to [0057], [0073] to [0089]; Fig. 4 (Family: none)	4-12, 20-22, 27-34, 42-44

 Further documents are listed in the continuation of Box C. See patent family annex.

"A"	Special categories of cited documents: document defining the general state of the art which is not considered to be of particular relevance	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"E"	earlier document but published on or after the international filing date	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"O"	document referring to an oral disclosure, use, exhibition or other means	"&"	document member of the same patent family
"P"	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search
24 October, 2003 (24.10.03)Date of mailing of the international search report
11 November, 2003 (11.11.03)Name and mailing address of the ISA/
Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP03/11679

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	JP 10-327381 A (Toshiba Corp.), 08 December, 1998 (08.12.98), Par. Nos. [0004] to [0017], [0043] to [0048], [0069] to [0092] (Family: none)	13-22, 35-44
P,A	JP 2003-249057 A (Toshiba Corp.), 05 September, 2003 (05.09.03), Full text (Family: none)	1-46
P,A	JP 2003-23604 A (LG Electronics Inc.), 24 January, 2003 (24.01.03), Full text (Family: none)	1-46
A	JP 11-191282 A (Toshiba Corp., Toshiba A.V.E. Kabushiki Kaisha), 13 July, 1999 (13.07.99), Full text & WO 99/34601 A2 & US 2001/0008576 A1	1-46

A. 発明の属する分野の分類 (国際特許分類 (IPC))

Int. C1' G11B27/00, G11B27/10, G11B20/10, G11B20/12

B. 調査を行った分野

調査を行った最小限資料 (国際特許分類 (IPC))

Int. C1' G11B27/00 - 27/06, G11B27/10, G11B20/10,
G11B20/12

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報 1922-1996年

日本国公開実用新案公報 1971-2003年

日本国実用新案登録公報 1996-2003年

日本国登録実用新案公報 1994-2003年

国際調査で使用した電子データベース (データベースの名称、調査に使用した用語)

C. 関連すると認められる文献

引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求の範囲の番号
X	J P 2000-57746 A (株式会社東芝, 東芝エー・ブイ・イー株式会社) 2000. 02. 25, 段落番号【0030】-【0088】, 【0210】-【0246】，第12-18図 & US 2001/0040862 A1	1-2, 23-25, 45-47
Y		4-22, 27-44
A		3, 26

 C欄の続きにも文献が列挙されている。 パテントファミリーに関する別紙を参照。

* 引用文献のカテゴリー

- 「A」特に関連のある文献ではなく、一般的技術水準を示すもの
- 「E」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの
- 「L」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献（理由を付す）
- 「O」口頭による開示、使用、展示等に言及する文献
- 「P」国際出願日前で、かつ優先権の主張の基礎となる出願

の日の後に公表された文献

- 「T」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの
- 「X」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの
- 「Y」特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの
- 「&」同一パテントファミリー文献

国際調査を完了した日 24. 10. 03	国際調査報告の発送日 11.11.03
国際調査機関の名称及びあて先 日本国特許庁 (ISA/JP) 郵便番号 100-8915 東京都千代田区霞が関三丁目4番3号	特許庁審査官 (権限のある職員) 宮下 誠 印 電話番号 03-3581-1101 内線 3590

C (続き) 関連すると認められる文献		
引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求の範囲の番号
Y	JP 11-161663 A (株式会社東芝) 1999. 06. 18, 段落番号【0018】-【0020】, 【0030】-【0031】 , 【0055】-【0057】, 【0073】-【0089】 , 第4図 (ファミリーなし)	4-12, 20-22, 27-34, 42-44
Y	JP 10-327381 A (株式会社東芝) 1998. 12. 08, 段落番号【0004】-【0017】, 【0043】-【0048】 , 【0069】-【0092】 (ファミリーなし)	13-22, 35-44
PA	JP 2003-249057 A (株式会社東芝) 2003. 09. 05, 全文 (ファミリーなし)	1-46
PA	JP 2003-23604 A (エルジー電子株式会社) 2003. 01. 24, 全文 (ファミリーなし)	1-46
A	JP 11-191282 A (株式会社東芝, 東芝エー・ブイ・イ 一株式会社) 1999. 07. 13, 全文 & WO 99/34 601 A2 & US 2001/0008576 A1	1-46